# Pattern Recognition

## (EE5907)

Song Bai

Email: songbai.site@gmail.com

# Outlines

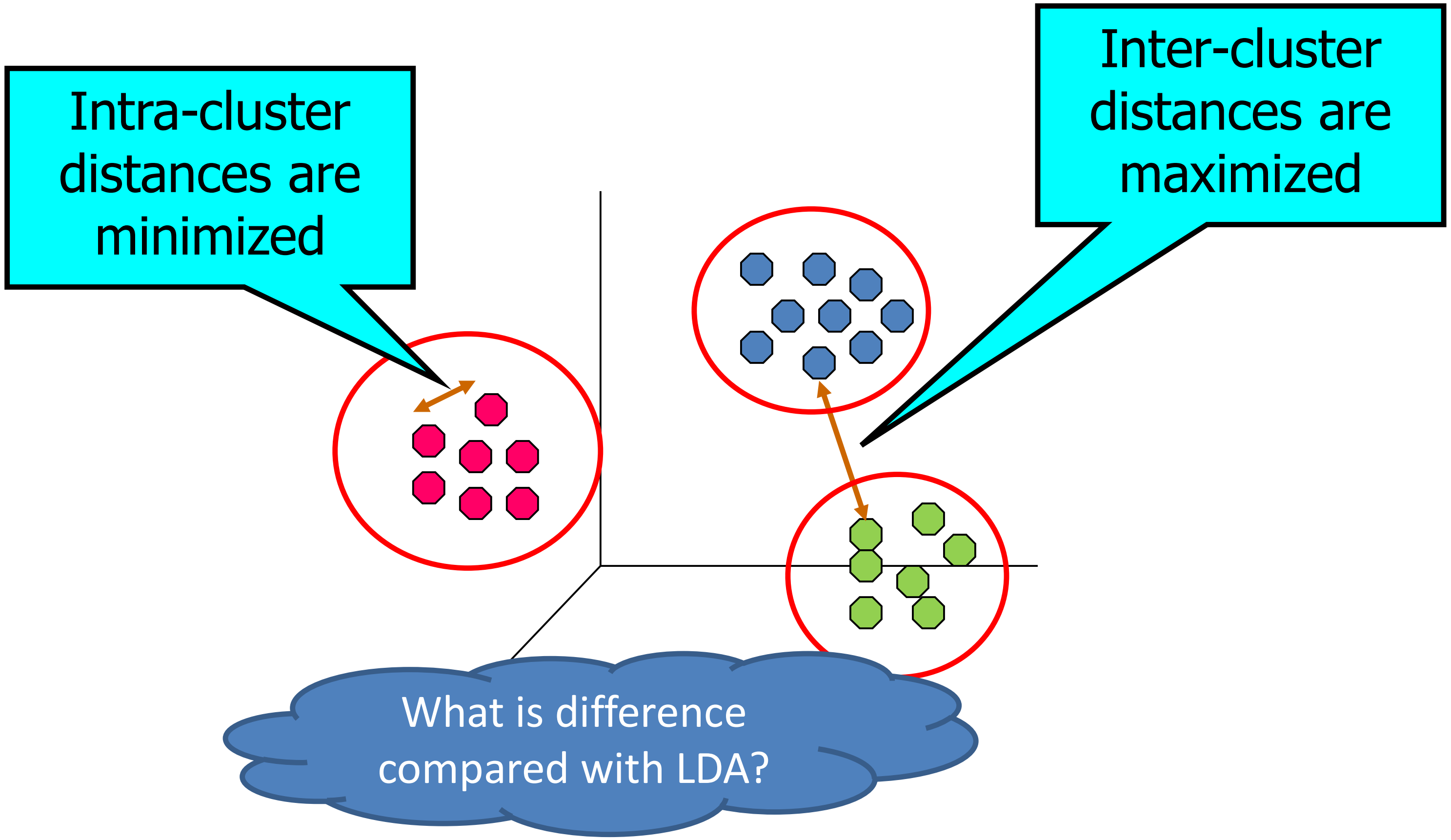
- Unsupervised Feature Extraction (PCA, NMF,…)
- Supervised Feature Extraction (LDA, GE, …)
- **Clustering and Applications**
- Gaussian Mixture Model
- Support Vector Machine
- Deep Learning

Some slides are from Shuicheng Yan, Henry Lin

# What is Cluster Analysis?

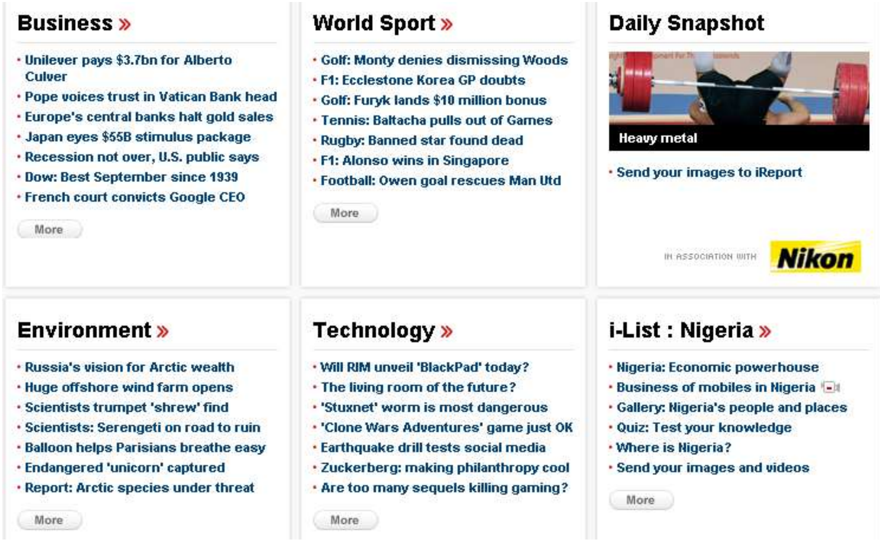
Implicit class label, not pre-defined!

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

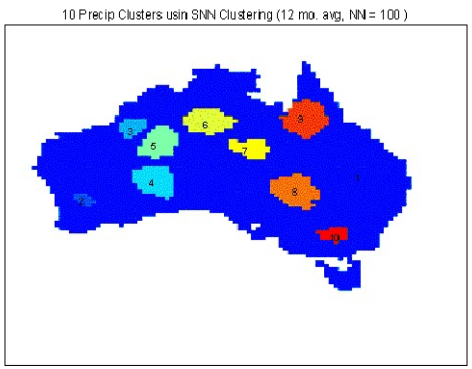


# Applications of Cluster Analysis

- Better understanding & search
  - Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations




- Visualization
  - Reduce the size of large data sets




Clustering rain fall amount in Australia

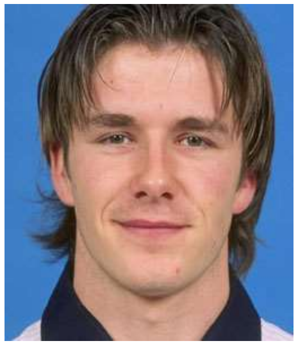
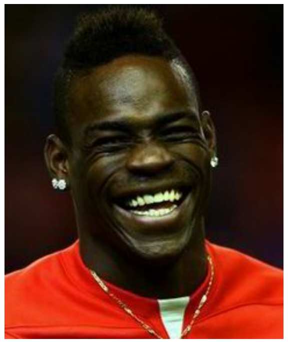
- Image Segmentation
  - Segment the image into regions

# What is not Cluster Analysis?

- Supervised classification
  - Have class label information


- Intuitive segmentation
  - Dividing students into different registration groups alphabetically, by first name

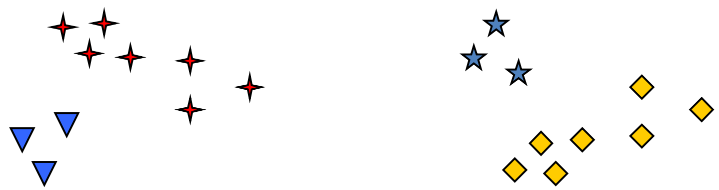


# Notion of a Cluster can be Ambiguous

How many clusters?

Two Clusters



Four Clusters

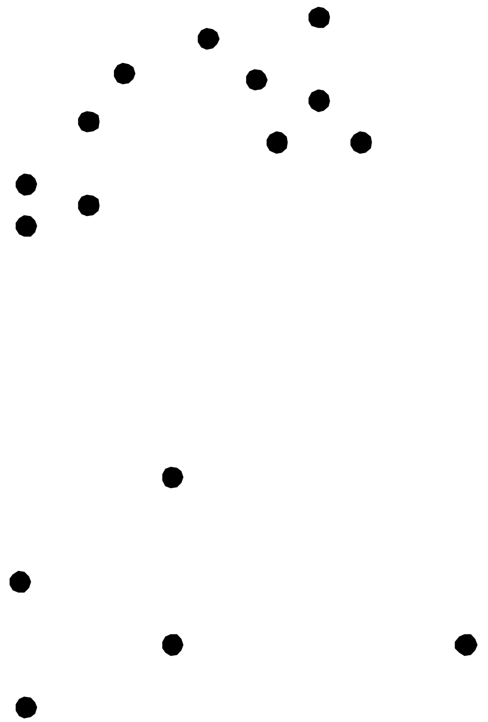
Six Clusters

# Types of Clustering

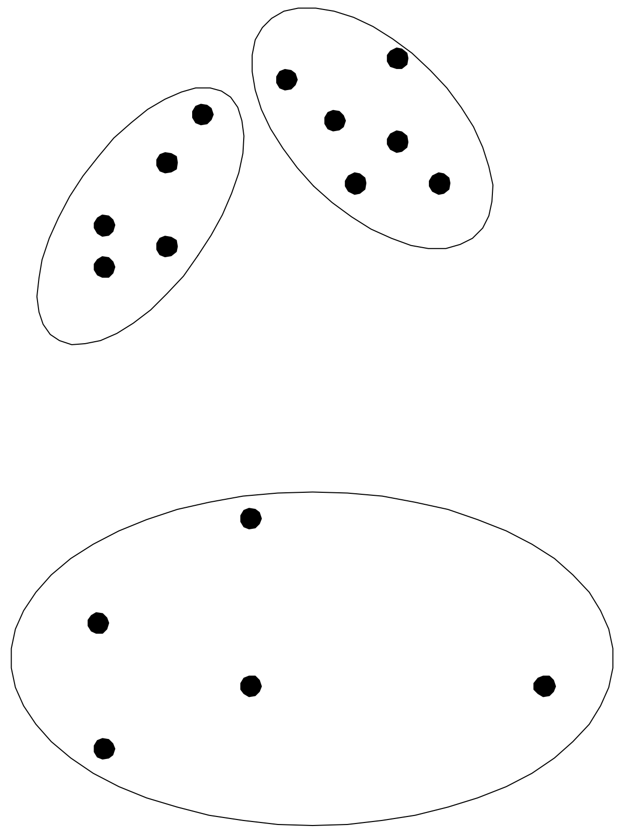
- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Cluster number is determined as you like

# Partitional Clustering

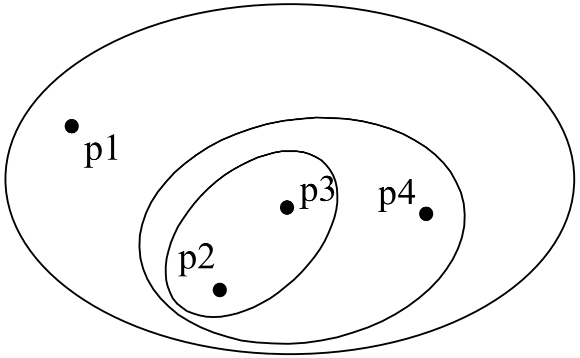


Original Points

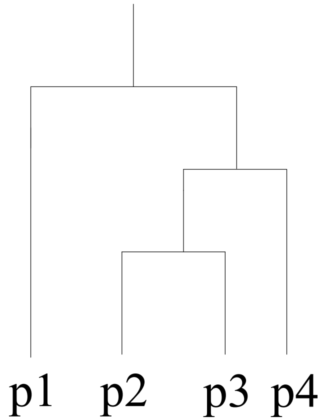


A Partitional  Clustering

# Hierarchical Clustering

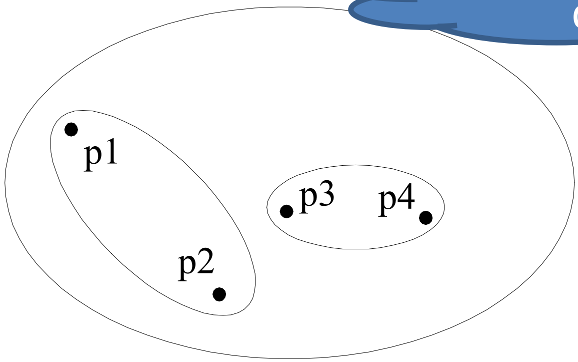



Traditional Hierarchical Clustering

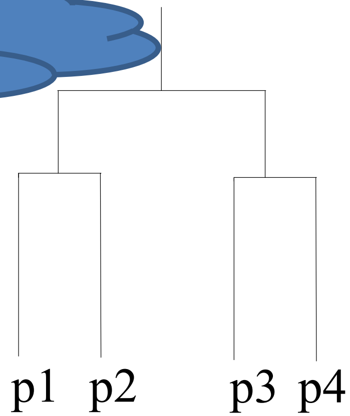



Traditional Dendrogram

What are the differences?

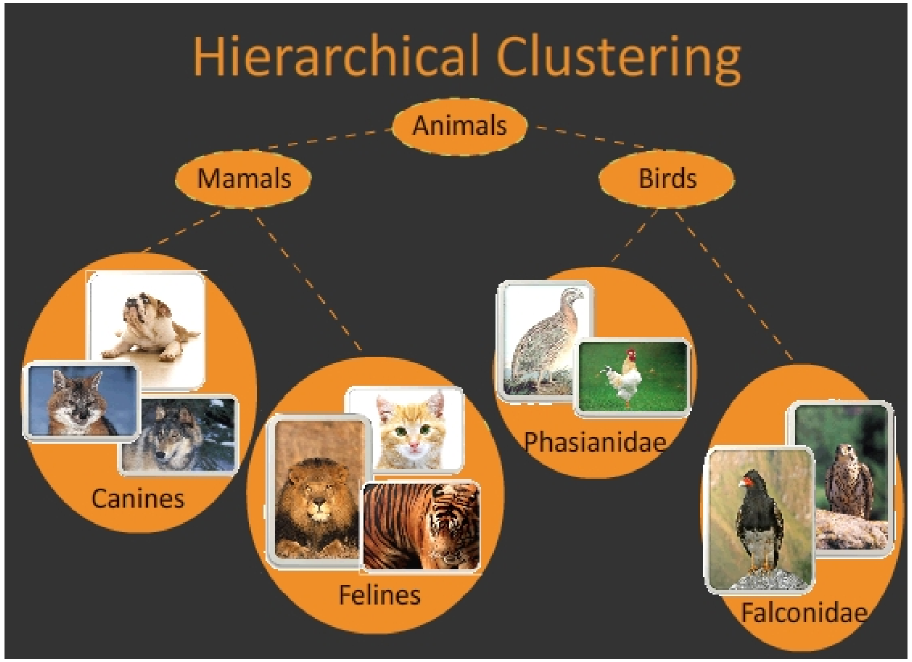



Non-traditional Hierarchical Clustering




Non-traditional Dendrogram

# Partitional Clustering vs. Hierarchical Clustering

## Other Distinctions Between Sets of Clusters

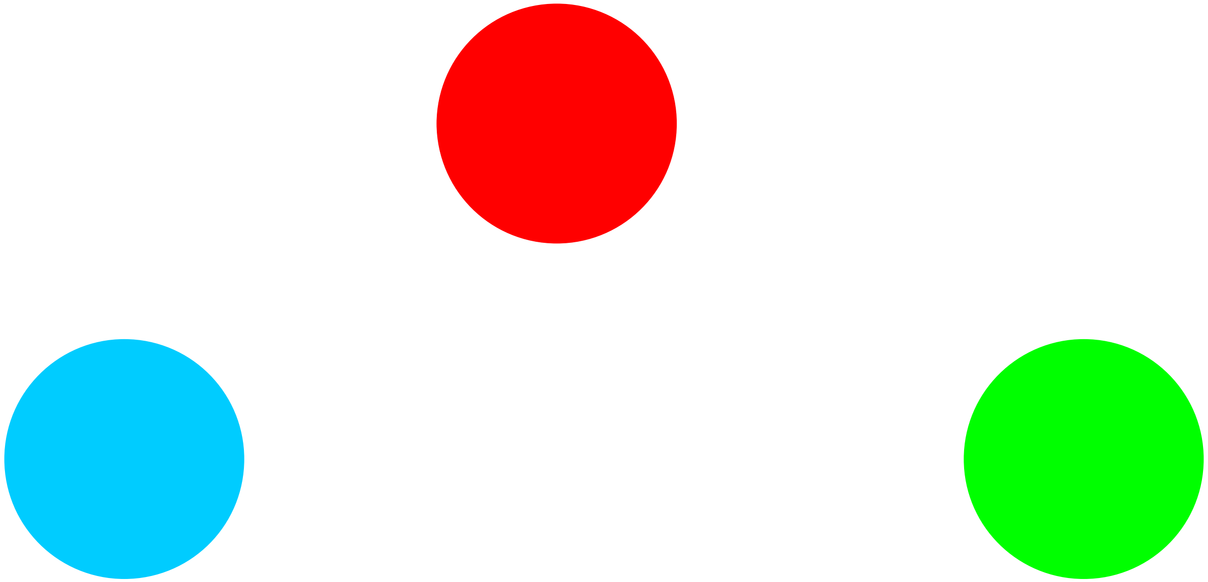
- Exclusive versus non-exclusive
  - In non-exclusive clustering, points may belong to multiple clusters.
  - Can represent multiple classes or 'border' points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics
- Partial versus complete
  - In some cases, we only want to cluster some of the data

# Types of Clusters

- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters

# Types of Clusters: Well-Separated

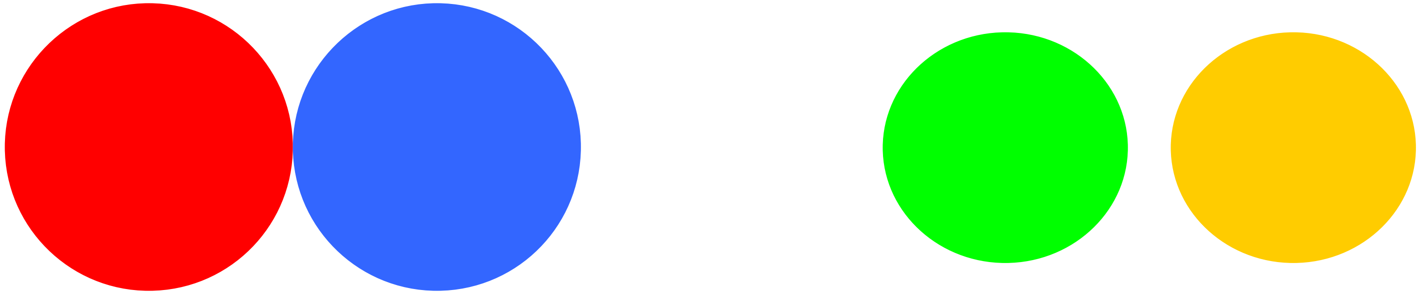
- Well-Separated Clusters:
  – A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

# Types of Clusters: Center-Based

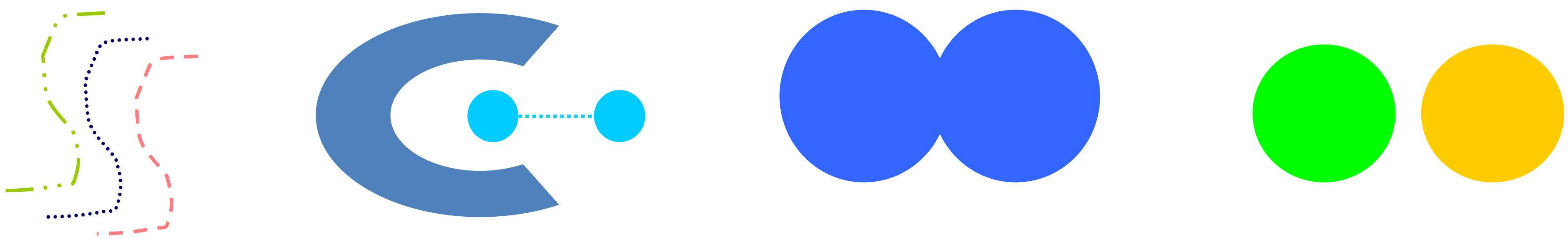
- Center-based
  - A cluster is a set of objects such that a point in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the average of all the points in the cluster, or the most "representative" point of a cluster



4 center-based clusters

# Types of Clusters: Contiguity-Based

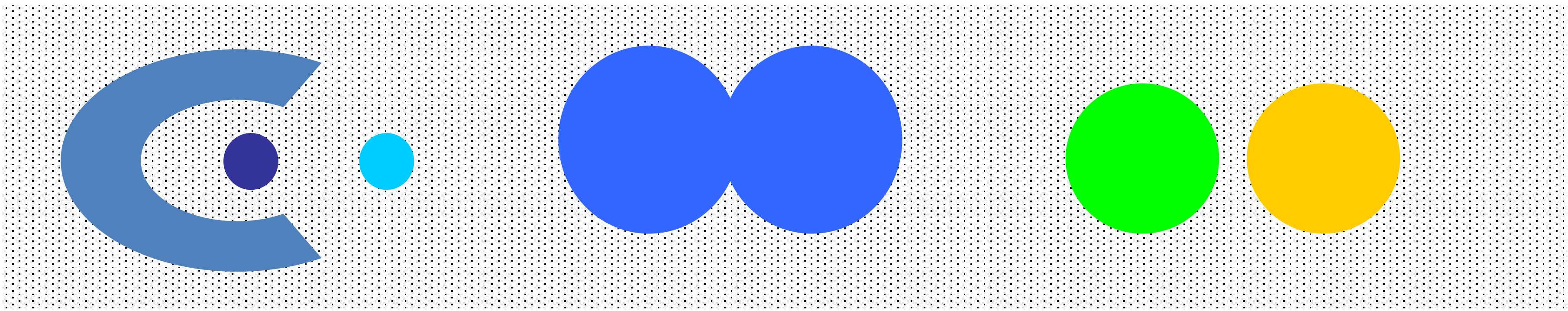
- Contiguous Cluster (Nearest neighbor)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to **one or more** other points in the cluster than to any point not in the cluster.



8 contiguous clusters

# Types of Clusters: Density-Based

- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when noise and outliers are present.



6 density-based clusters

# Clustering Algorithms

- K-means
- Hierarchical clustering

# K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is very simple

“How” is the key!
Discuss!

```
Select K points as the initial centroids.
repeat
    Form K clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don’t change
```

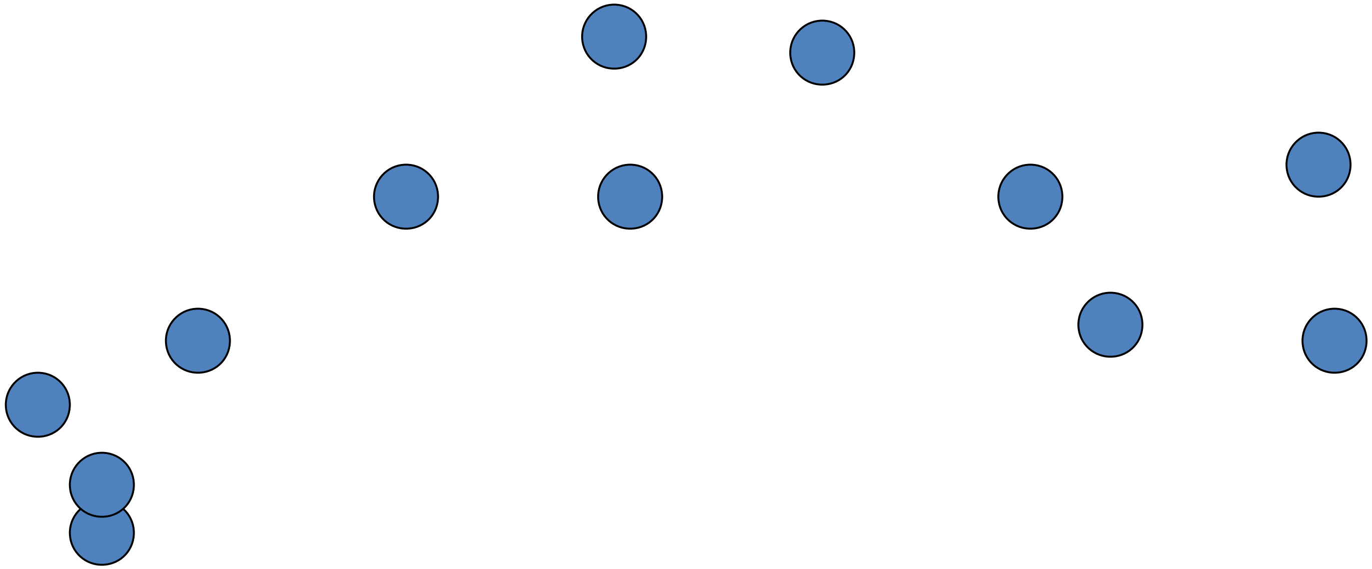
# K-means Clustering – Details

- 初始中心点通常是随机选择的。
所产生的群组因不同的运行而不同

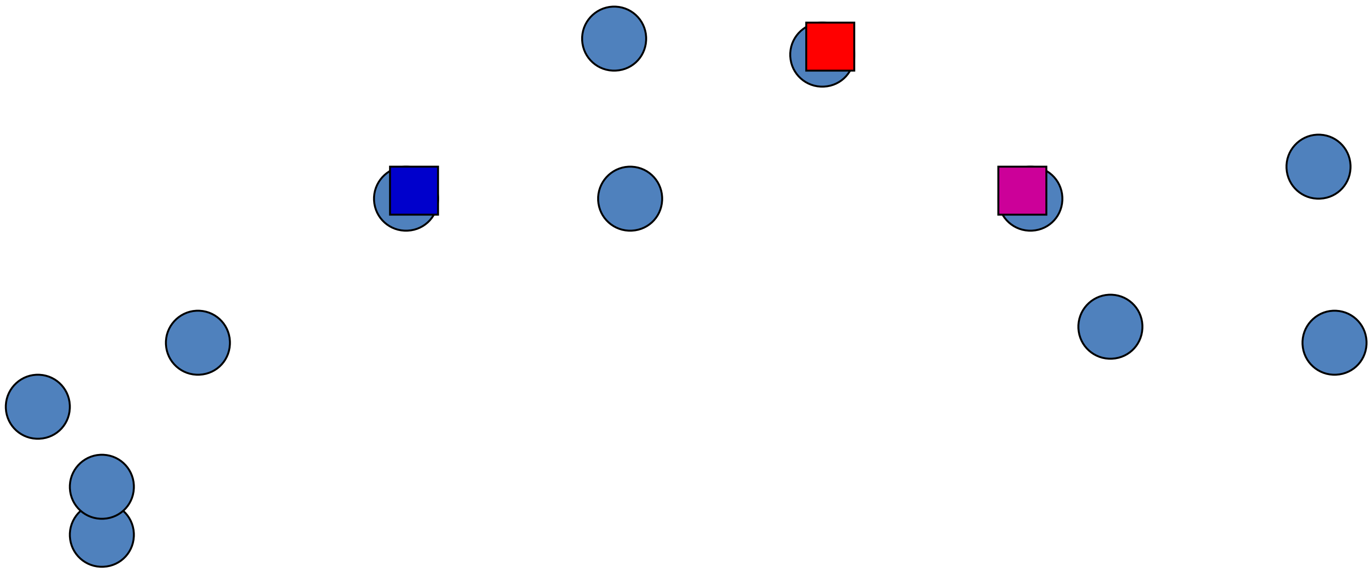
- Initial centroids are often chosen **randomly**.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- 'Closeness' is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to 'Until relatively few points change clusters'
- Complexity is O( n * K * I * d )
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of features

- 中心点是（通常）聚类中各点的平均值。
- "紧密性 "由欧氏距离、余弦相似度、相关性等来衡量
- K-means将收敛上述常见的相似度。
- 大多数收敛发生在最初的几次迭代中。 - 通常停止条件被改为 "直到相对较少的点改变集群" - 复杂性为O( n * K * I * d ) - n =点的数量，K =集群的数量，I =迭代的数量，d =特征的数量

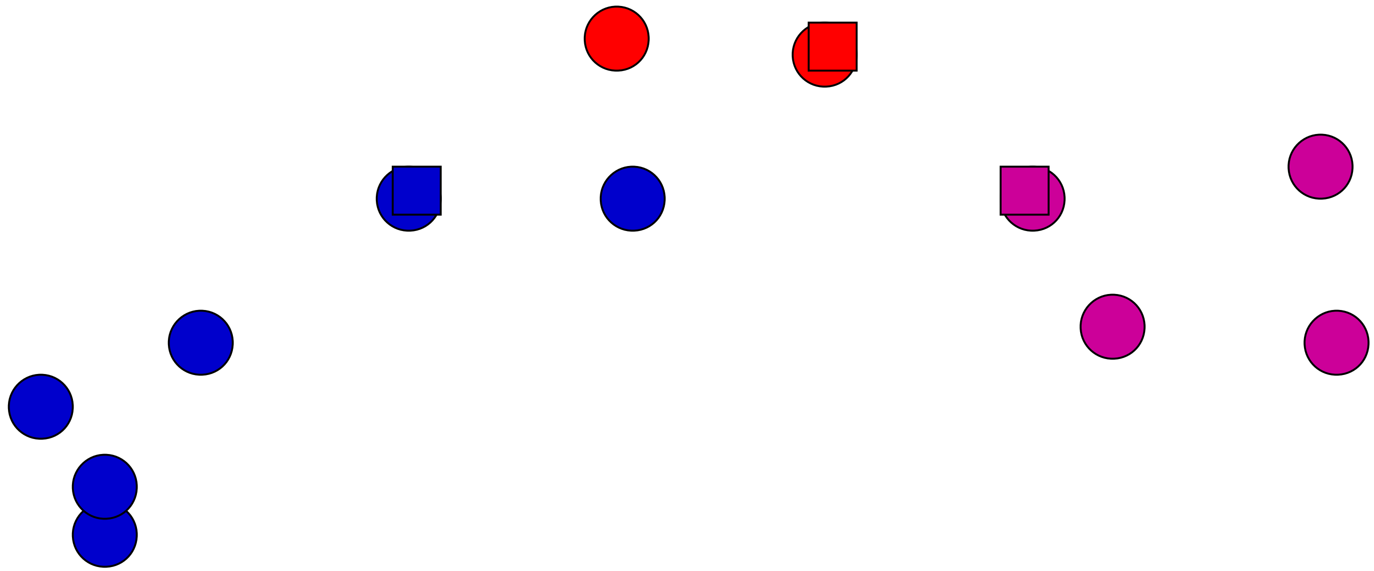
# K-means: an example



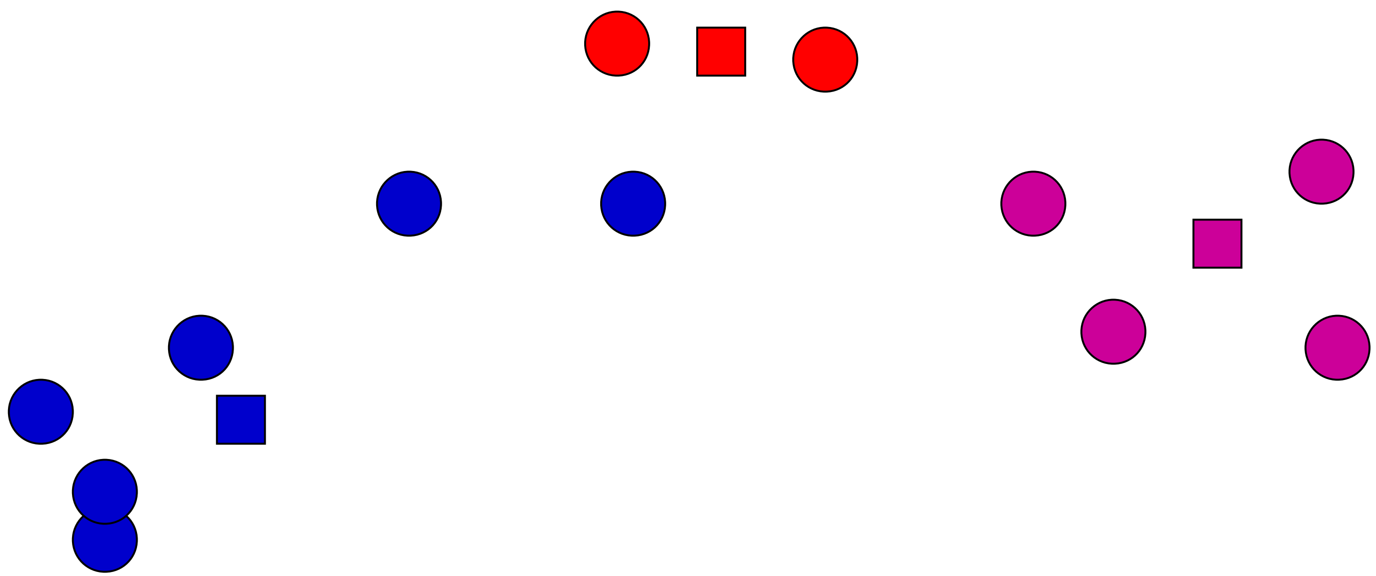
# K-means: Initialize centers randomly



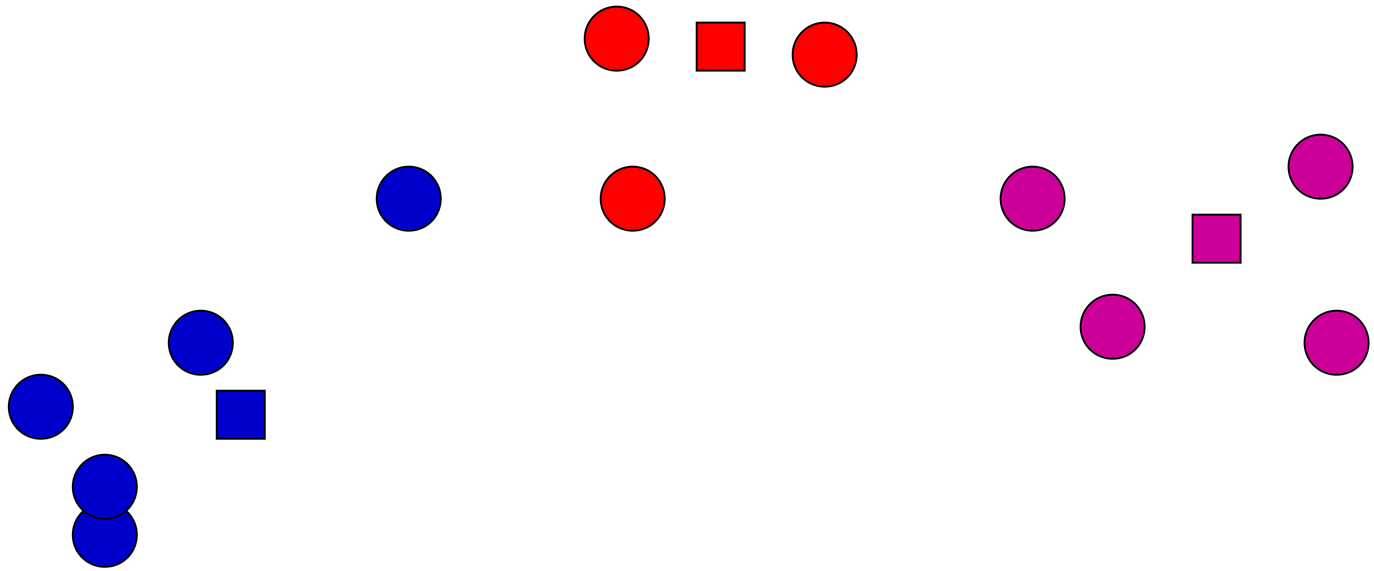
# K-means: assign points to nearest center



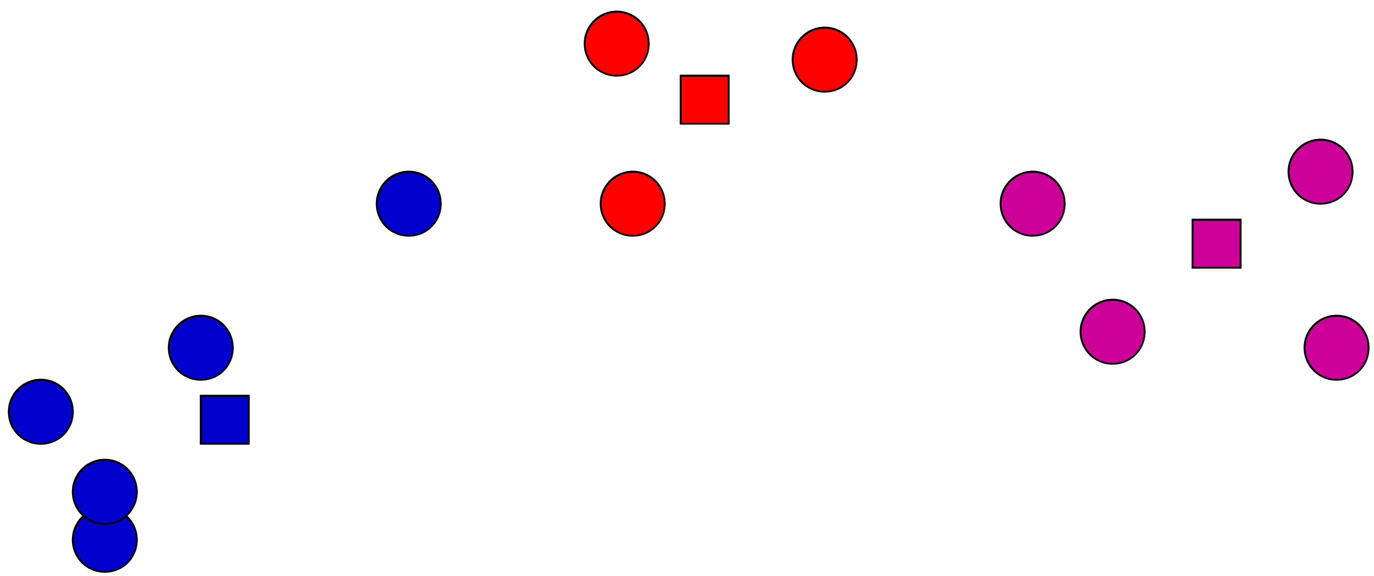
# K-means: readjust centers



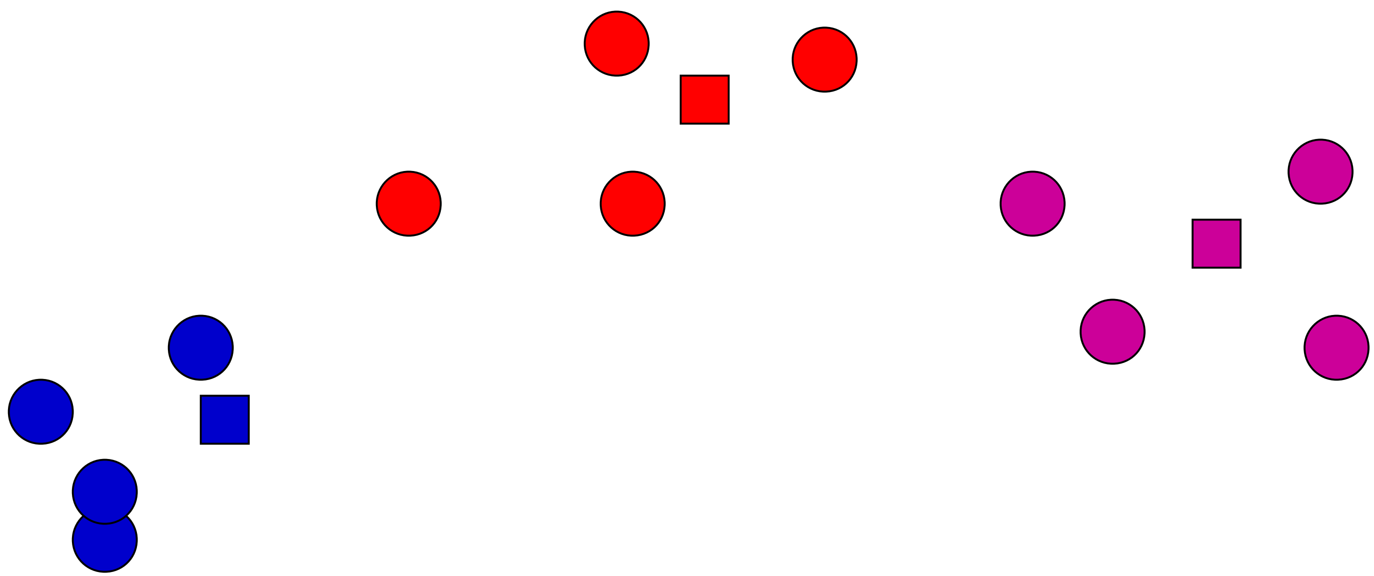
# K-means: assign points to nearest center



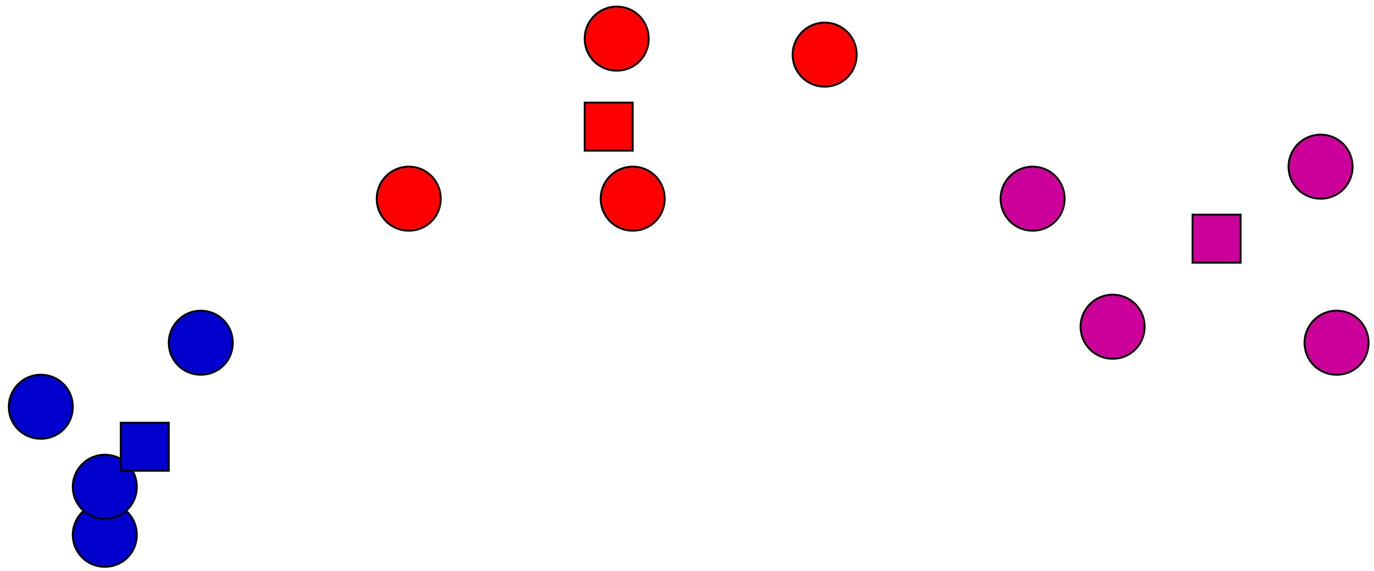
# K-means: readjust centers



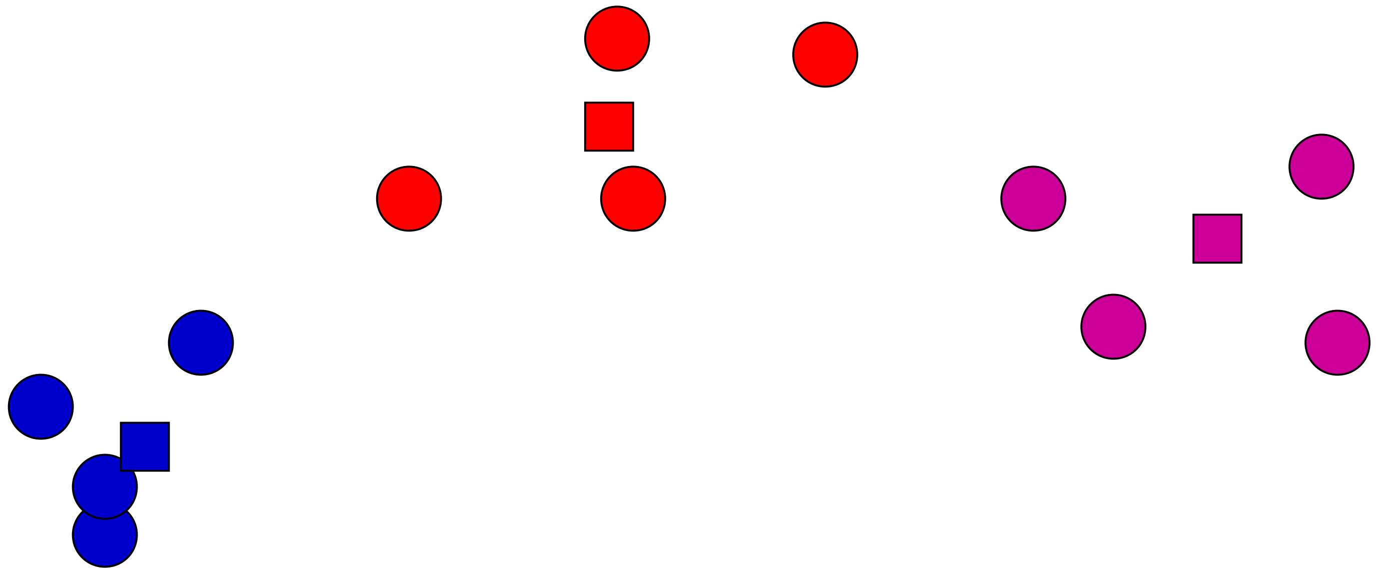
# K-means: assign points to nearest center



# K-means: readjust centers

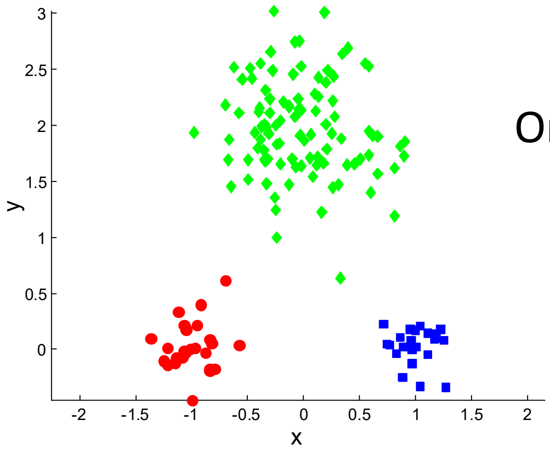


# K-means: assign points to nearest center

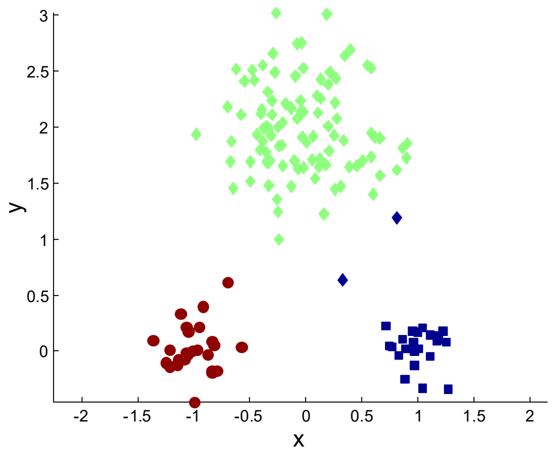


No changes: Done

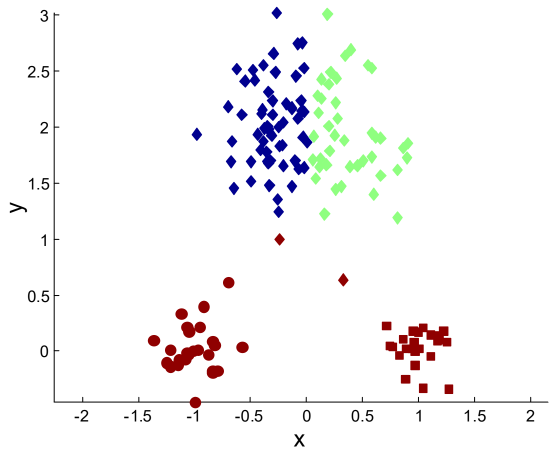
# Two different K-means Clusterings




Original Points




Optimal Clustering




Sub-optimal Clustering

# Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them:

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

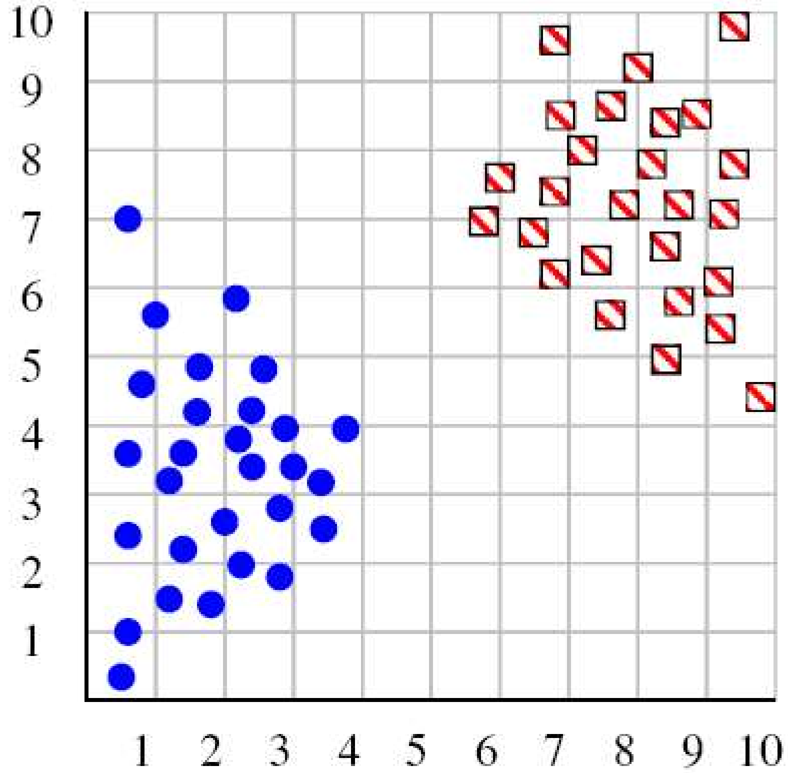
  - $x$ is a data point in cluster $C_i$ and $m_i$ is the representative point for cluster $C_i$
    - $m_i$ corresponds to the center (mean) of the cluster mostly
  - Given many clusterings, we can choose the one with the smallest error
  - One easy way to reduce SSE is to increase K, the number of clusters
    - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

- 最常见的衡量标准是平方误差之和（SSE）。
- 对于每一个点，误差是与最近的集群的距离
- 为了得到SSE，我们将这些误差平方，然后求和。

- x是聚类Ci中的一个数据点，mi是聚类Ci的代表点
- mi主要对应于群集的中心（平均值）。
- 给出许多聚类，我们可以选择误差最小的那一个
- 减少SSE的一个简单方法是增加K，即聚类的数量
- 一个好的聚类，其K值较小，比一个差的聚类，其SSE值较高。

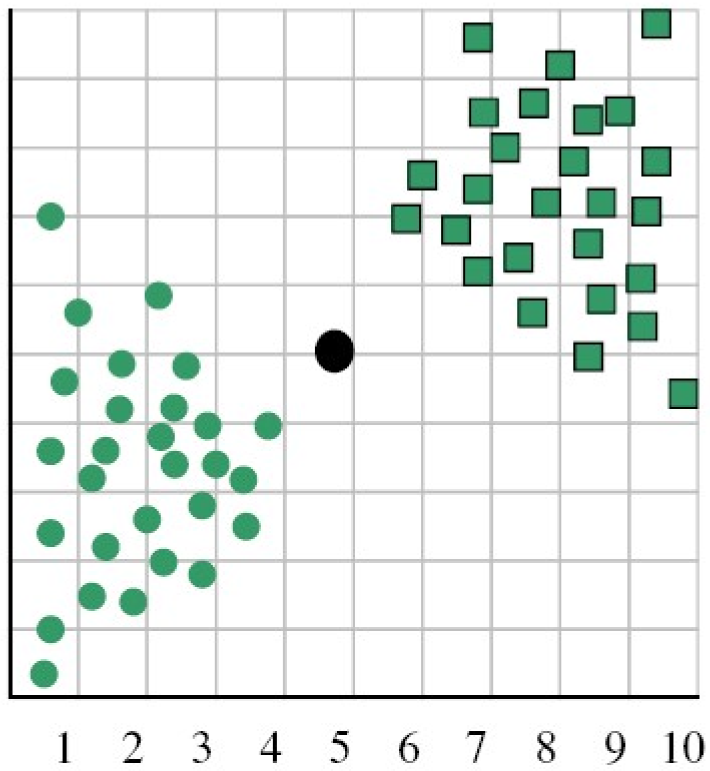
# Deciding K

- Try different Ks



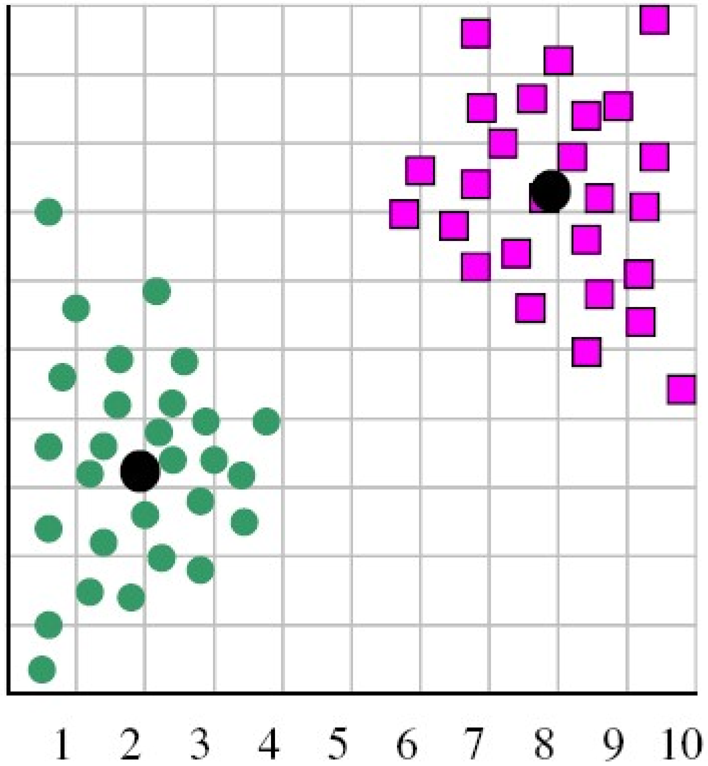
# Deciding K

- When K = 1, SSE = 873.0



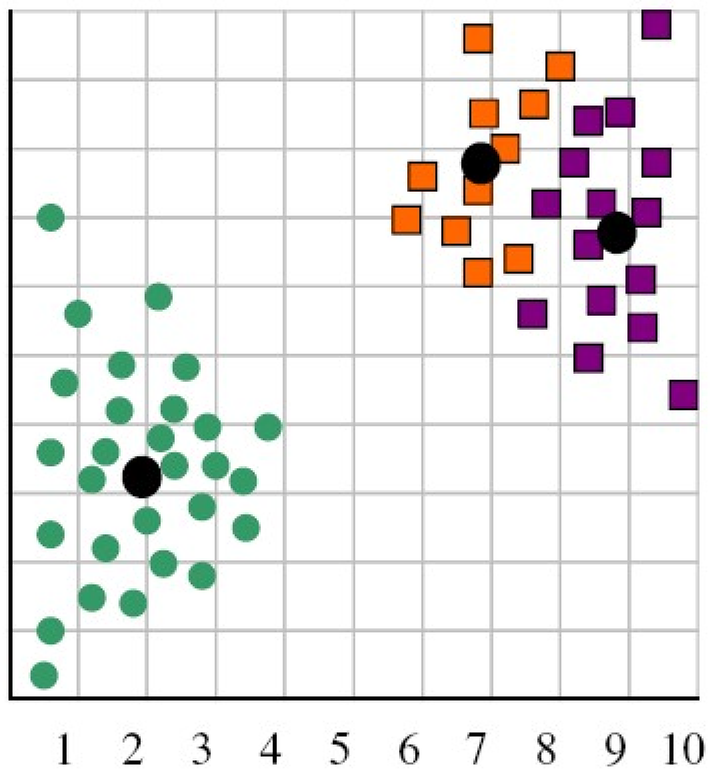
# Deciding K

- When K = 2, SSE = 173.1



# Deciding K

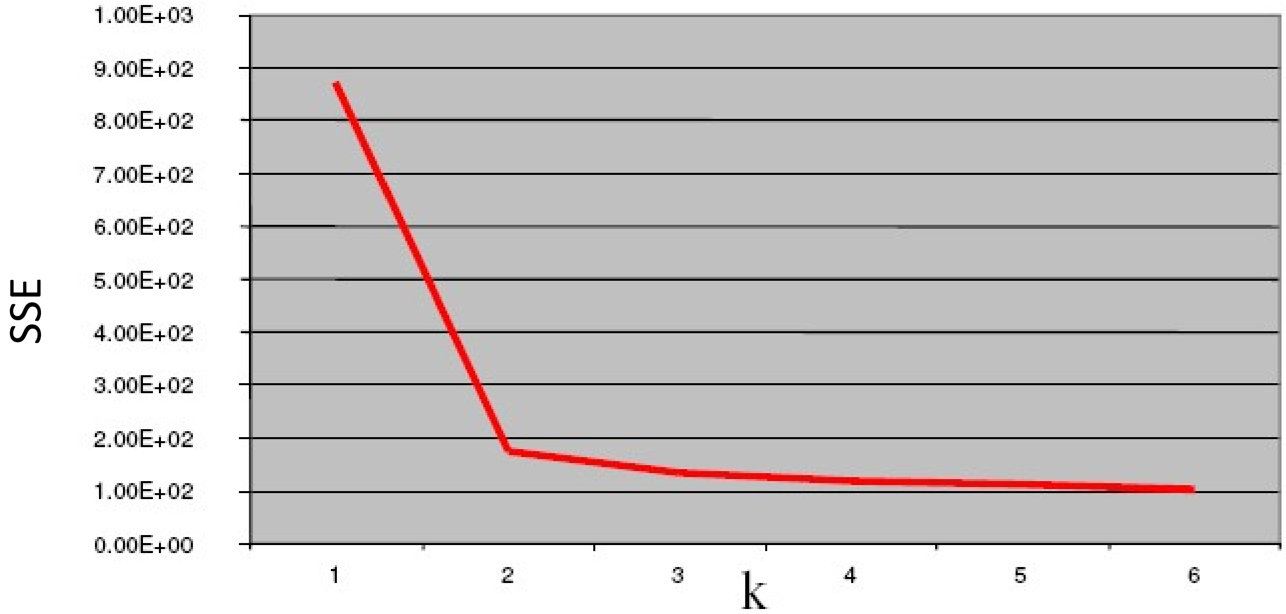
- When K = 3, SSE = 133.6



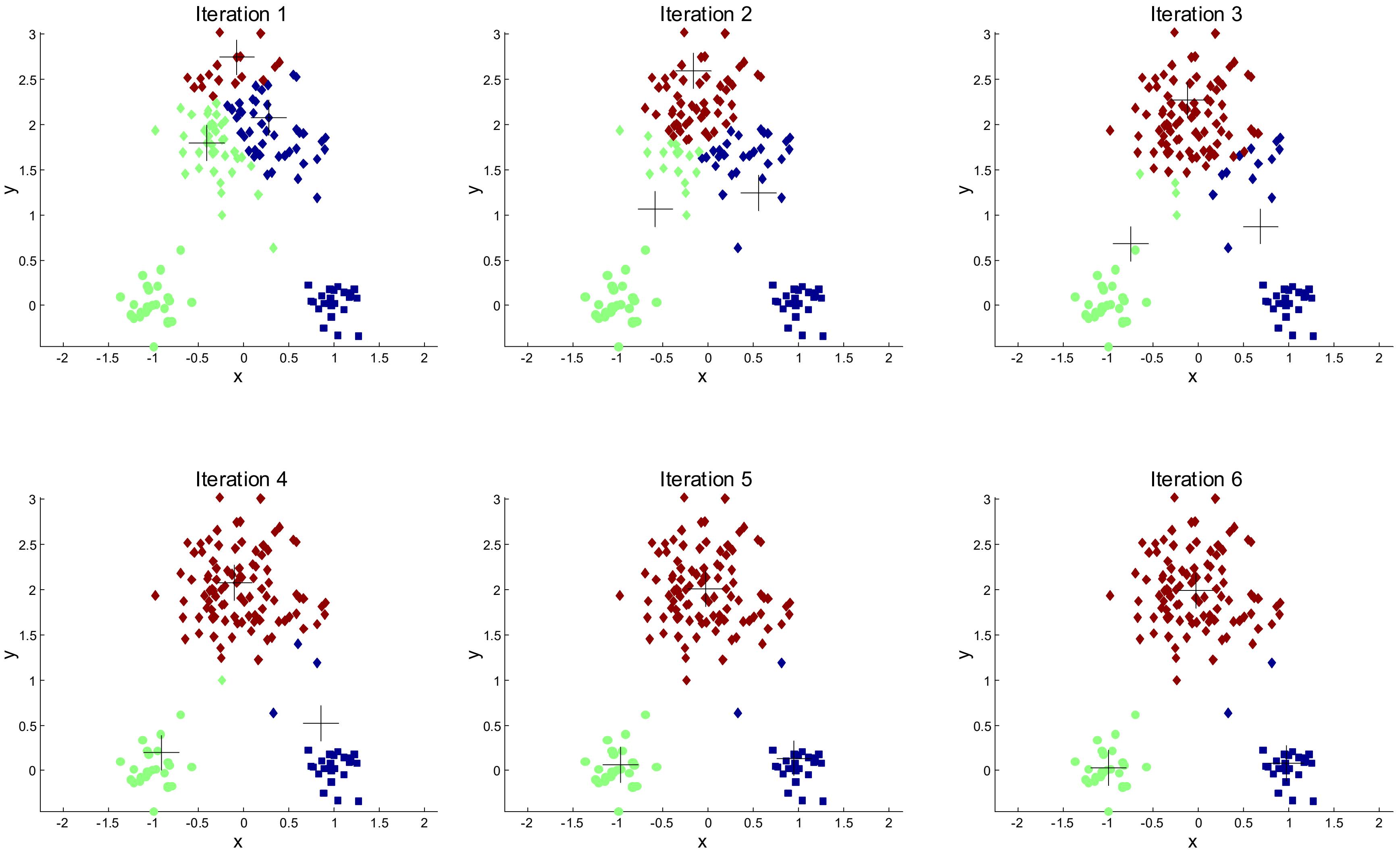
# Deciding K

- 我们可以绘制K=1到6的目标函数值
- K=2时的突然变化高度提示了两个集群
- "寻找膝盖 "或 "寻找肘部"
- 请注意，结果并不总是像这个玩具例子中那样清晰明了

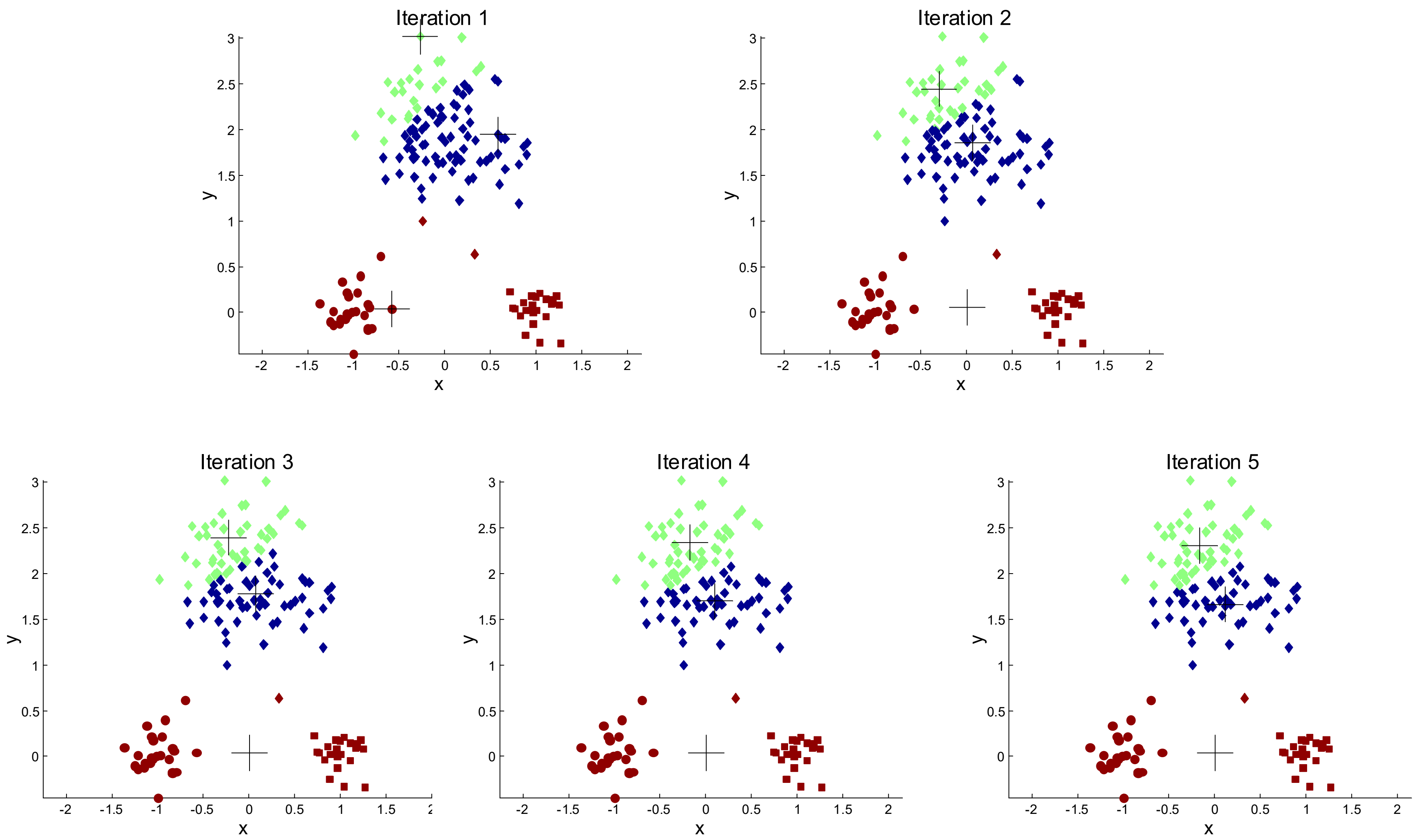
- We can plot objective function values for K=1 to 6
- The abrupt change at K=2 is highly suggestive of two clusters
- “knee finding” or “elbow finding”
- Note that the results are not always as clear cut as in this toy example



# Importance of Choosing Initial Centroids



# Importance of Choosing Initial Centroids …



# Problems with Selecting Initial Points

- If there are K ‘real’ clusters then the chance of selecting one centroid from each cluster is small.
  – Chance is relatively small when K is large
  – Sometimes the initial centroids will readjust themselves in ‘right’ way, and sometimes they don’t

- 如果有K个 "真正的 "聚类，那么从每个聚类中选择一个中心点的机会就很小。
- 当K大的时候，机会就相对较小
- 有时初始中心点会以 "正确 "的方式重新调整，有时则不会。

# Solutions to Initial Centroids Problem

- 多次运行
- 有帮助，但概率不在你这边
- 使用分层聚类法来确定初始中心点
- 选择K个以上的初始中心点，然后在这些初始中心点中进行选择
- 选择相距最远的
- 后期处理

- Multiple runs
  - Helps, but probability is not on your side
- Use hierarchical clustering to determine initial centroids
- Select more than K initial centroids and then select among these initial centroids
  - Select most widely separated
- Postprocessing

# Post-processing

- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE

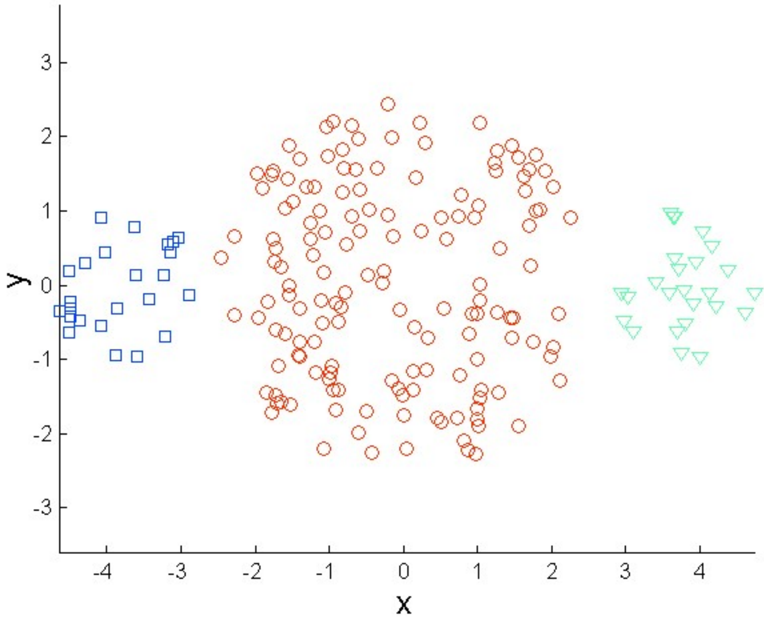
- 后期处理
- 消除可能代表离群值的小聚类
- 分割 "松散 "的聚类，即具有相对较高SSE的聚类
- 合并 "接近 "且SSE相对较低的聚类

# Limitations of K-means

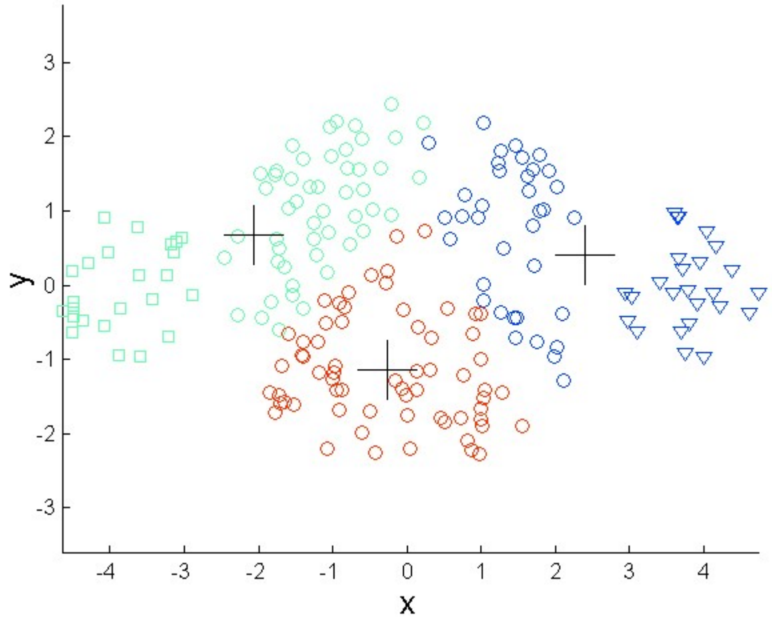
- 当集群的大小不同时，K-means有问题。
- 大小
- 密度
- 当数据中含有异常值（不属于任何群组）时，K-means会出现问题。
- 相似性函数是否合适。

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
- K-means has problems when the data contains outliers (not belonging to any cluster).
- The similarity function is suitable or not.

# Limitations of K-means: Differing Sizes

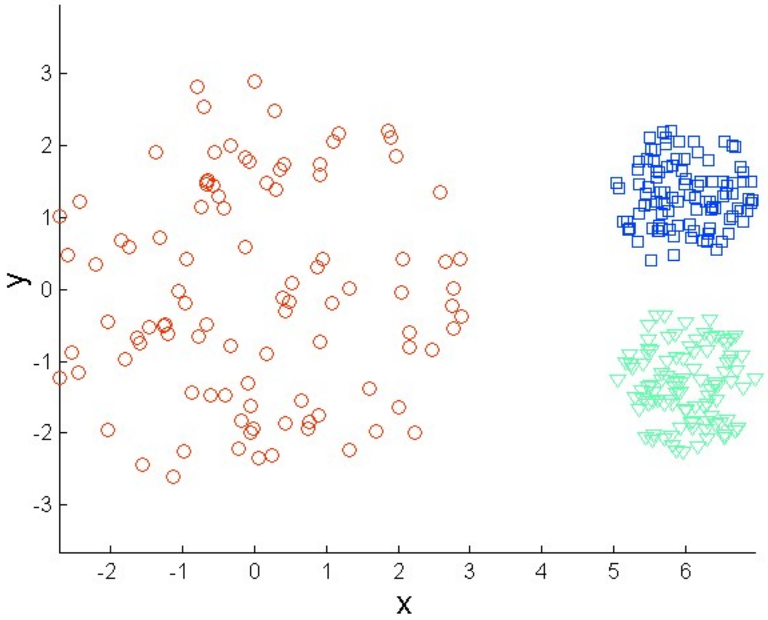



Original Points

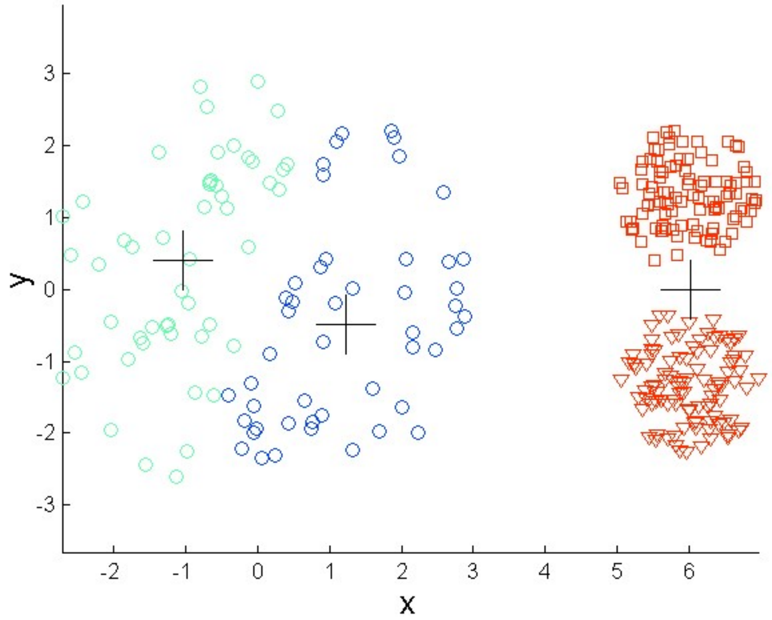



K-means (3 Clusters)

# Limitations of K-means: Differing Density

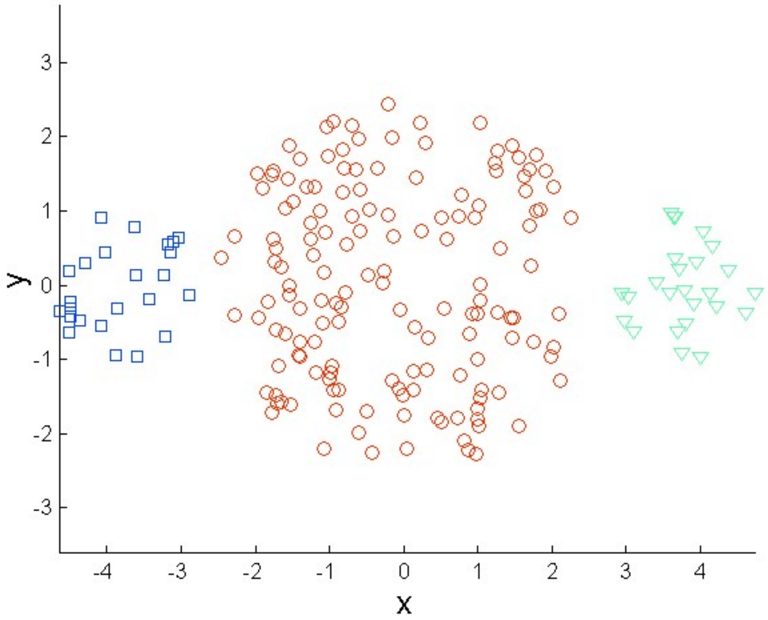



Original Points

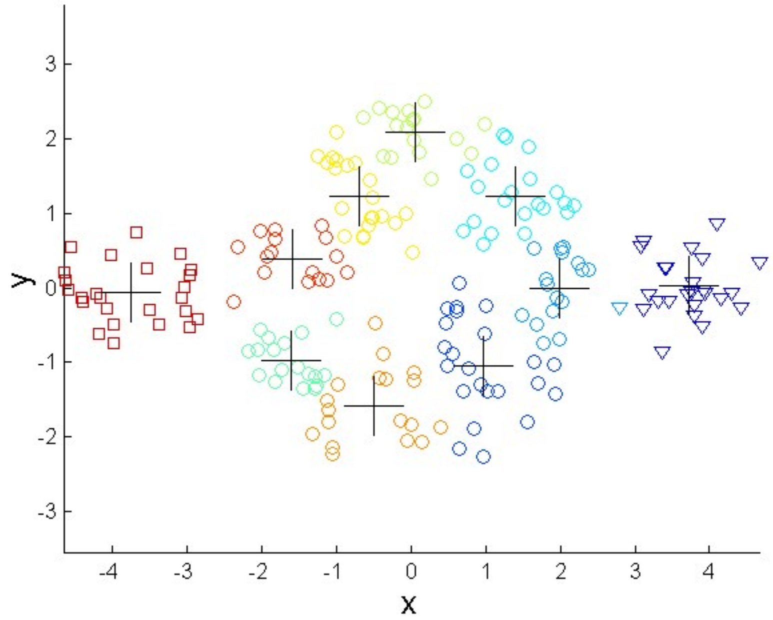



K-means (3 Clusters)

# Overcoming K-means Limitations




Original Points



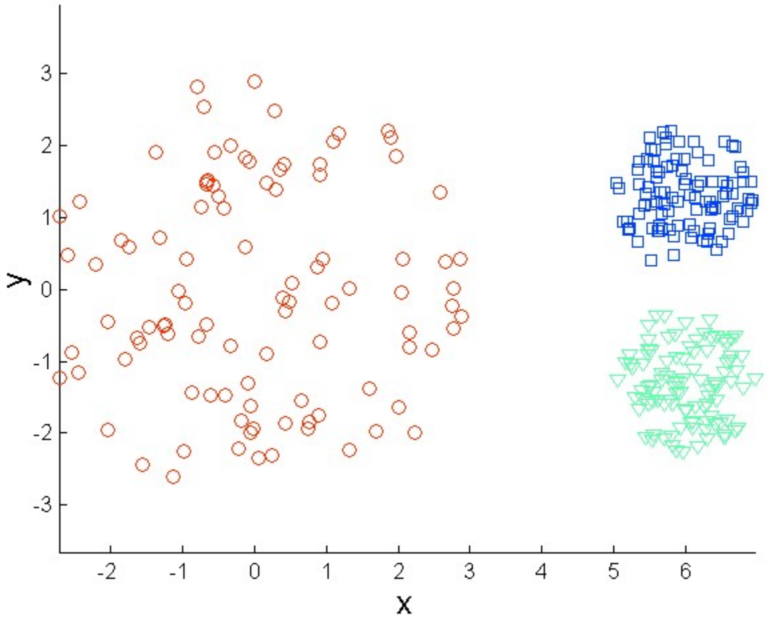

K-means Clusters

One solution is to use many clusters.

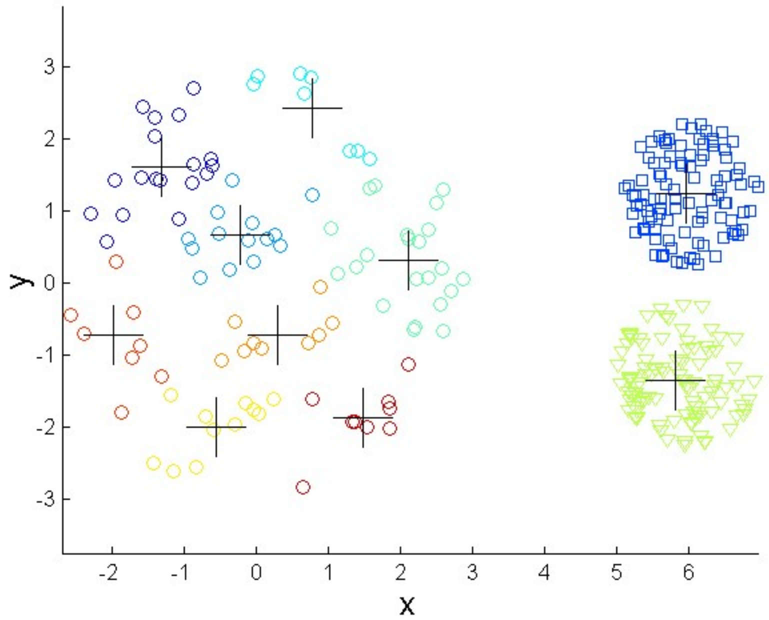
Find parts of clusters, but need to put together.

找到集群的部分，但需要把它们放在一起。

# Overcoming K-means Limitations




Original Points




K-means Clusters

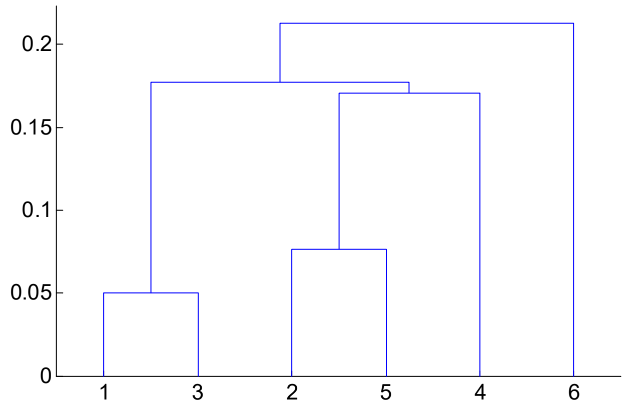
# Example

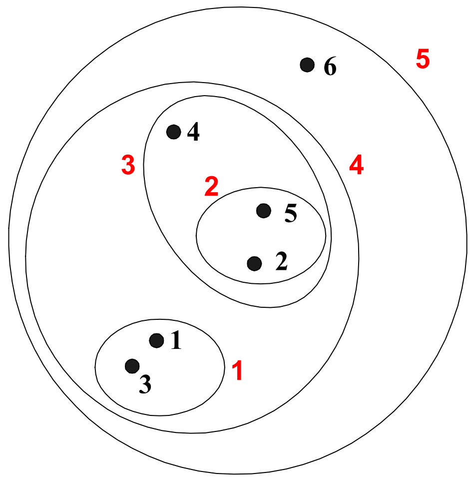
Four data points are $x_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $x_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $x_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $x_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and the task is to partition the data into two clusters. i) Perform k-means with two initial centroids as $\{x_1, x_2\}$, please list the iterative centroids until converged; and ii) perform k-means with two initial centroids as $\{x_2, x_4\}$, please list the iterative centroids until converged.

# Hierarchical Clustering 分层聚类

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequence of merges or splits

- 产生一组嵌套的聚类，组织成一棵分层的树
- 可以可视化为树状图
- 一个树状图，记录了合并或拆分的顺序



# Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by 'cutting' the dendrogram at the proper level

- 不需要假定任何特定数量的聚类
- 通过在适当的层次上 "切割 "树状图，可以得到任何所需数量的聚类。

# Hierarchical Clustering

- 分层聚类的两种主要类型
- 聚合型。
- 从各点作为单独的聚类开始
- 在每一步，合并最接近的一对聚类，直到只剩下一个聚类（或k个聚类）。
- 分裂式。
- 从一个包罗万象的聚类开始
- 在每一步，分割一个簇，直到每个簇包含一个点（或有k个簇）。
- 传统的分层算法使用相似度或距离矩阵
- 每次合并或拆分一个聚类

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

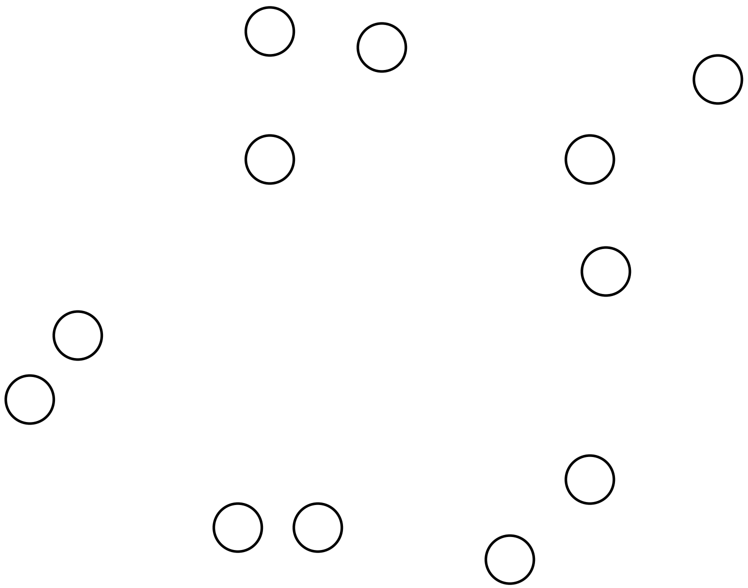
# Agglomerative Clustering Algorithm

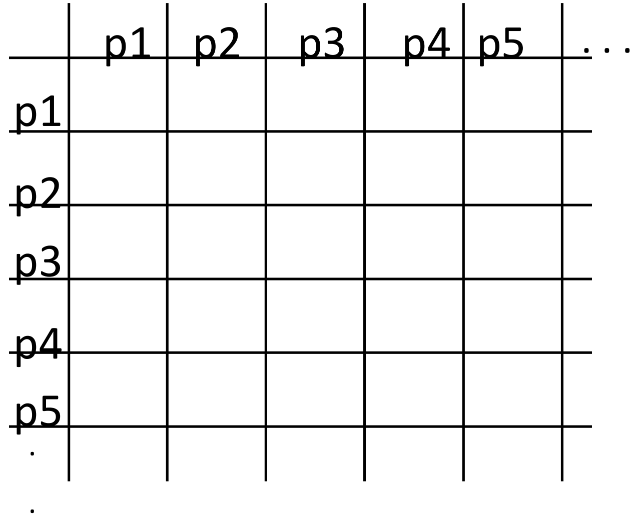
- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  1. Compute the proximity/distance matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the proximity/distance matrix
  6. **Until** only a single cluster remains
- Key operation is the computation of the proximity of two clusters
  – Different approaches to defining the distance between clusters distinguish the different algorithms

- 比较流行的分层聚类技术
- 基本算法简单明了
1. 计算接近/距离矩阵
2. 让每个数据点成为一个聚类
3. 3.重复进行
4. 合并最接近的两个聚类
5. 更新接近/距离矩阵
6. 直到只剩下一个集群
- 关键操作是计算两个聚类的接近度
- 不同的算法有不同的方法来定义聚类之间的距离。

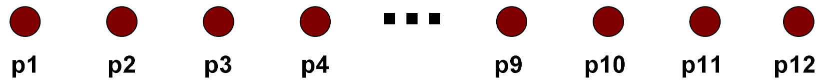
# Starting Situation

- Start with clusters of individual points and a proximity matrix



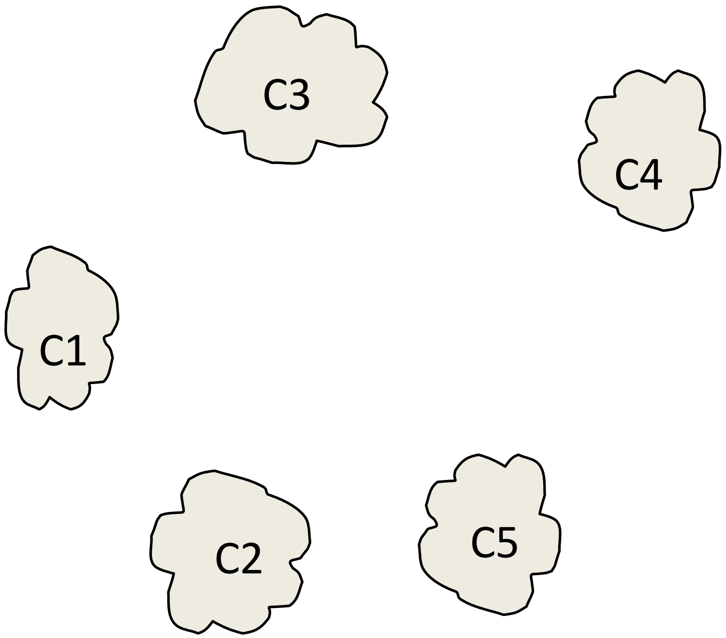



Proximity Matrix



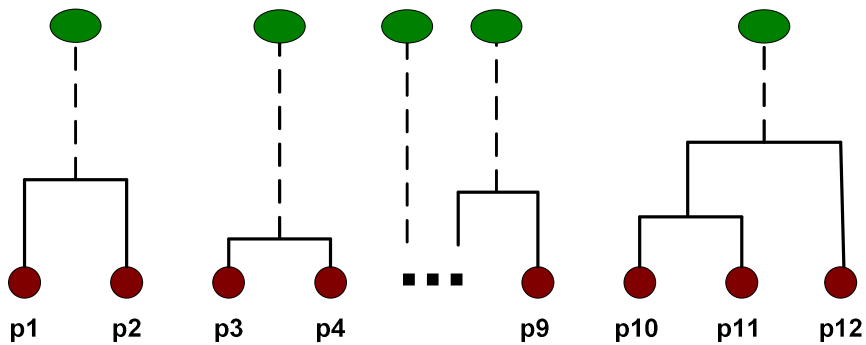
# Intermediate Situation

- After some merging steps, we have some clusters



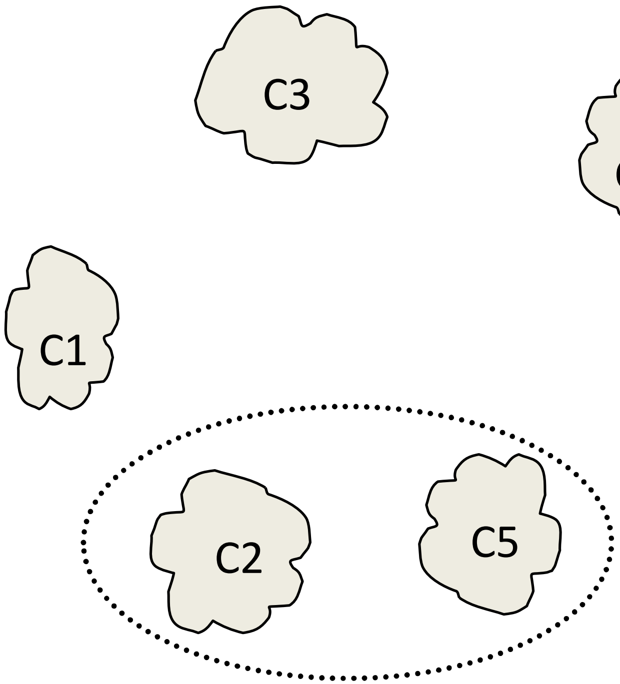

| | C1 | C2 | C3 | C4 | C5 |
|---|---|---|---|---|---|
| C1 | | | | | |
| C2 | | | | | |
| C3 | | | | | |
| C4 | | | | | |
| C5 | | | | | |

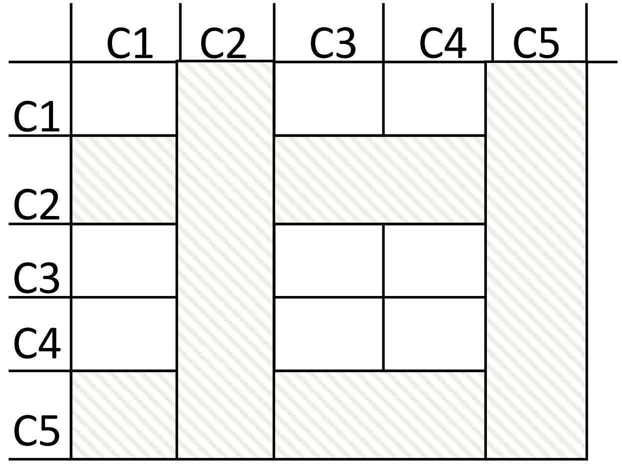
Proximity Matrix



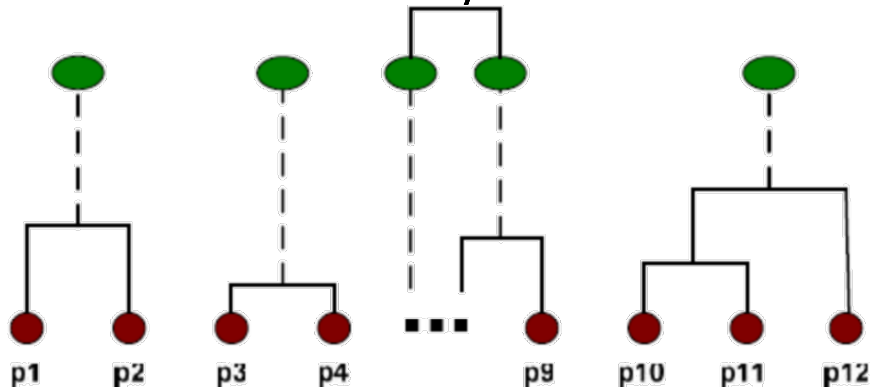
# Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



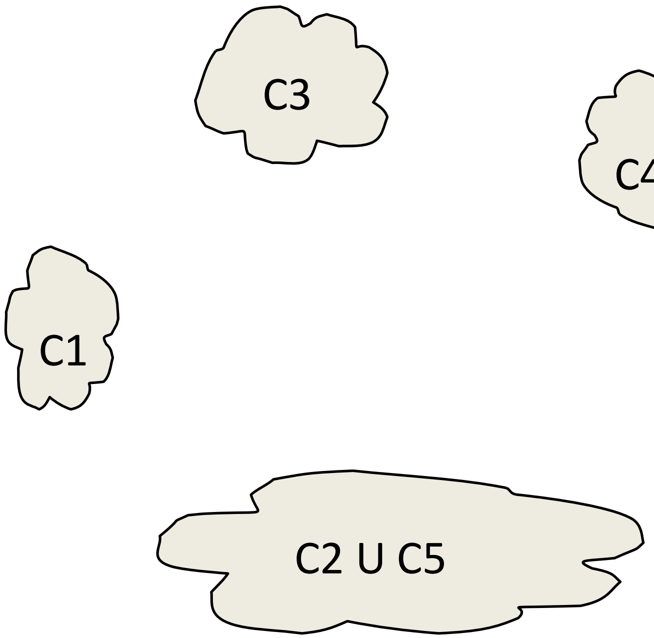




Proximity Matrix



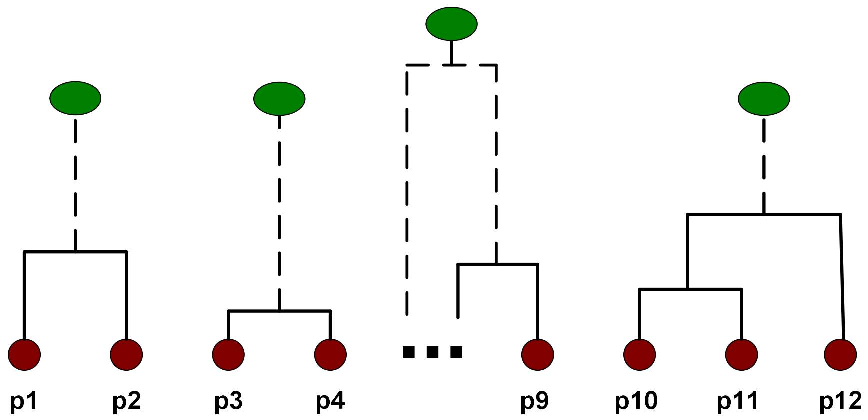
# After Merging

- The question is "How do we update the proximity matrix?"

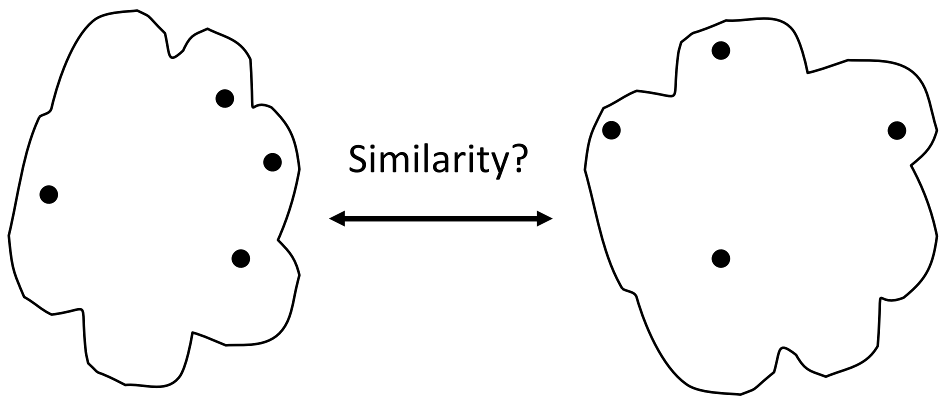



| | C1 | C2 U C5 | C3 | C4 |
|---|---|---|---|---|
| C1 | | ? | | |
| C2 U C5 | ? | ? | ? | ? |
| C3 | | ? | | |
| C4 | | ? | | |

Proximity Matrix



# How to Define Inter-Cluster Similarity

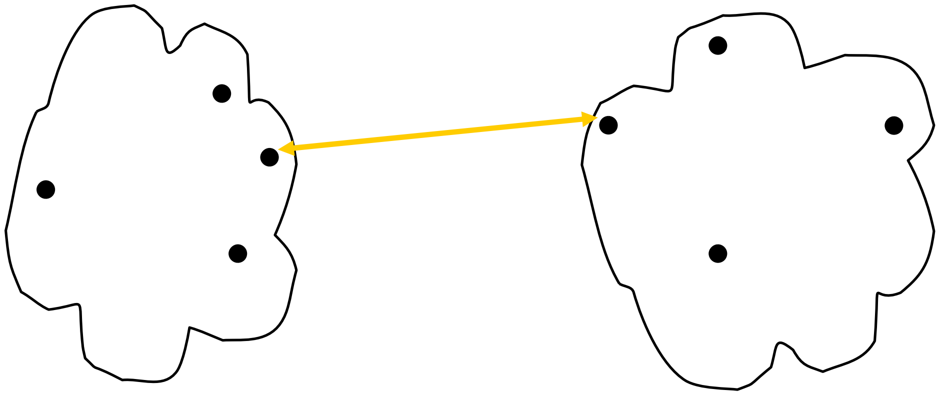



| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

# How to Define Inter-Cluster Similarity

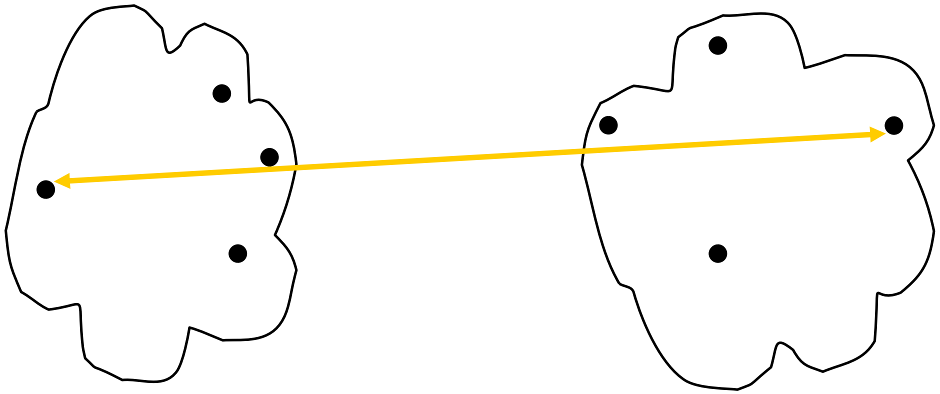


- MIN
- MAX
- Group Average
- Distance Between Centroids

| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

Proximity Matrix

# How to Define Inter-Cluster Similarity

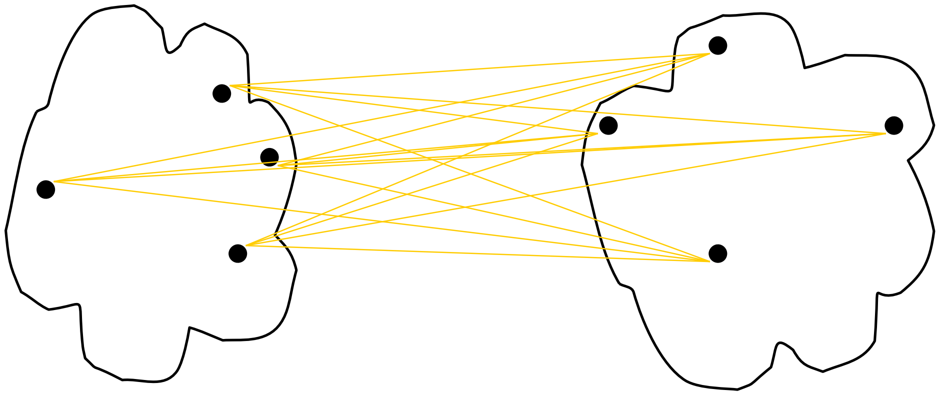


| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

# How to Define Inter-Cluster Similarity



| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |

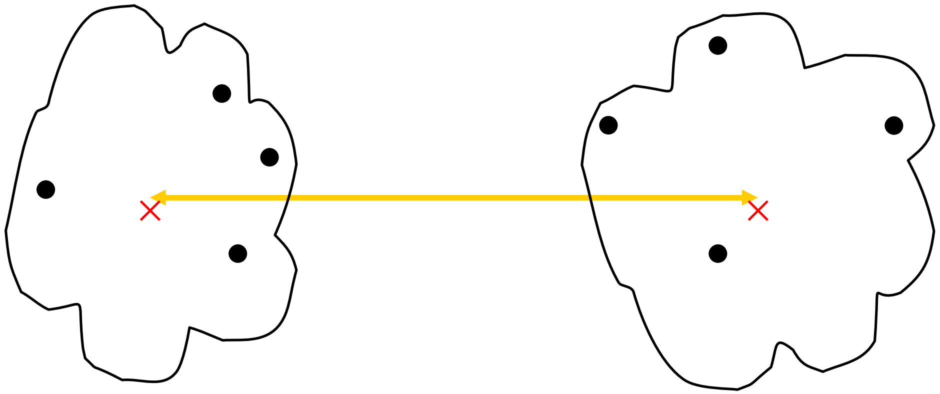
.

.

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

# How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids

| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

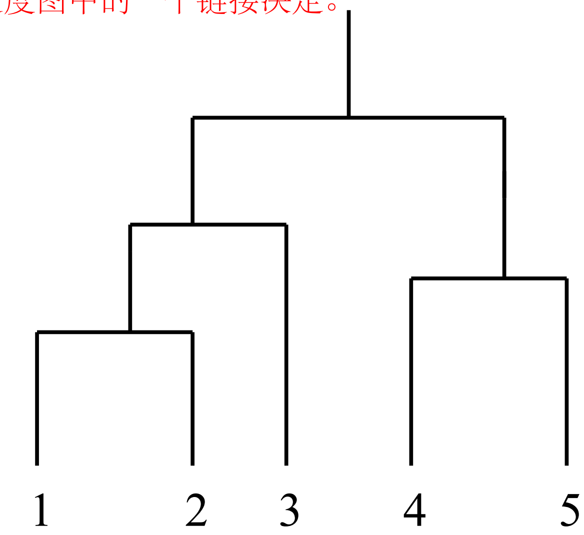
Proximity Matrix

# Cluster Similarity: MIN or Single Link

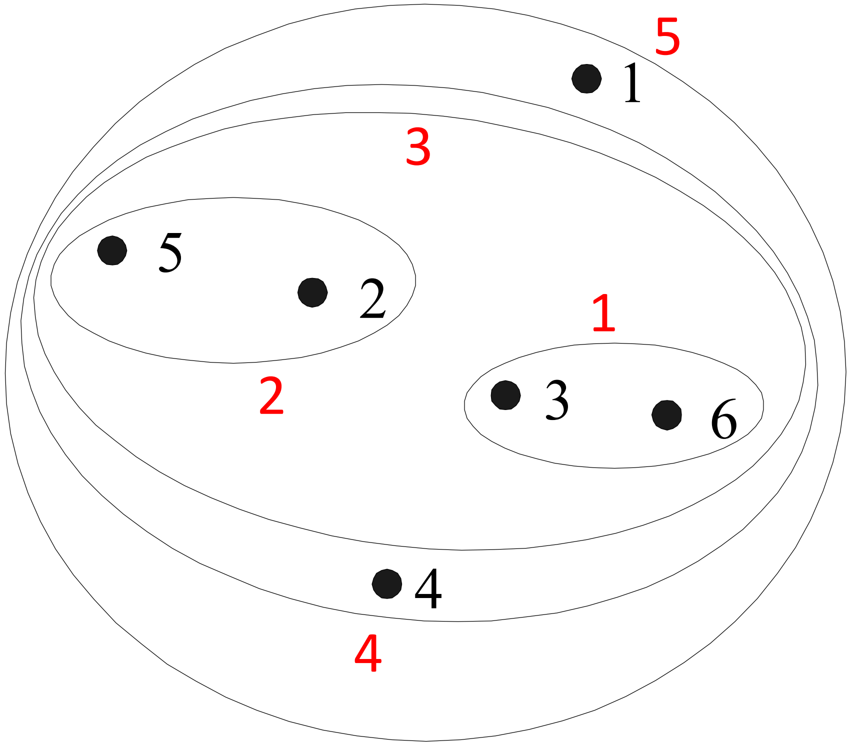
- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  – Determined by one pair of points, i.e., by one link in the proximity graph.

- 两个聚类的相似性是基于不同聚类中两个最相似（最接近）的点
- 由一对点决定，即由接近度图中的一个链接决定。

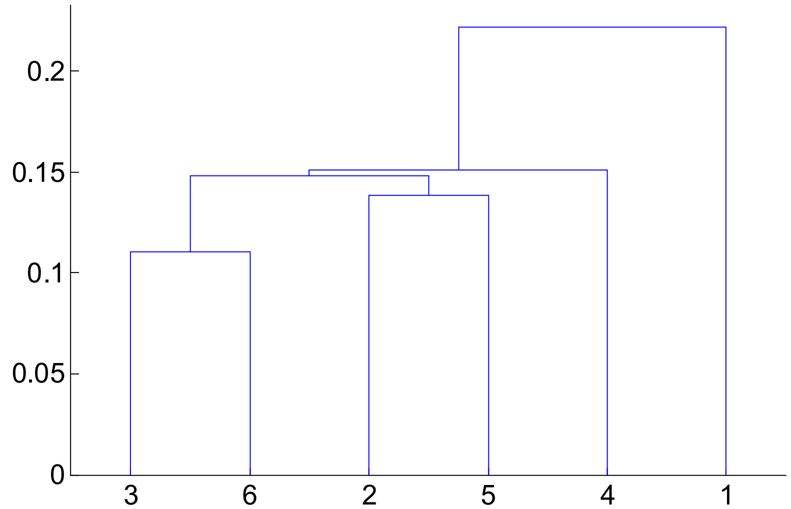
| | I1 | I2 | I3 | I4 | I5 |
|---|---|---|---|---|---|
| I1 | 1.00 | 0.90 | 0.10 | 0.65 | 0.20 |
| I2 | 0.90 | 1.00 | 0.70 | 0.60 | 0.50 |
| I3 | 0.10 | 0.70 | 1.00 | 0.40 | 0.30 |
| I4 | 0.65 | 0.60 | 0.40 | 1.00 | 0.80 |
| I5 | 0.20 | 0.50 | 0.30 | 0.80 | 1.00 |



# Hierarchical Clustering: MIN

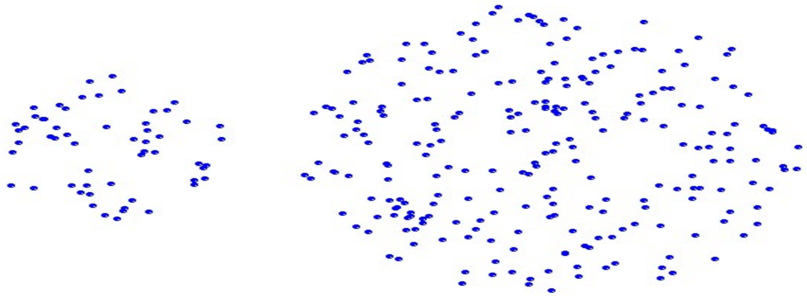



Nested Clusters

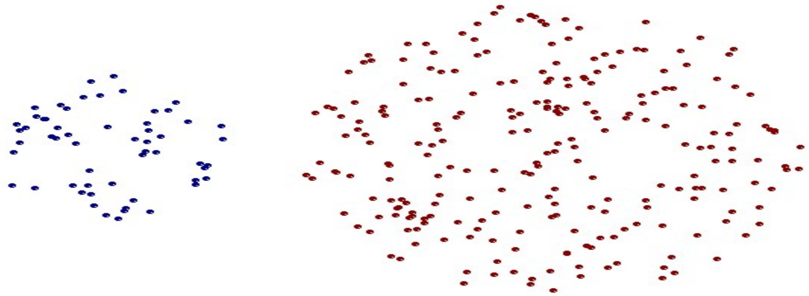



Dendrogram

# Strength of MIN



Original Points

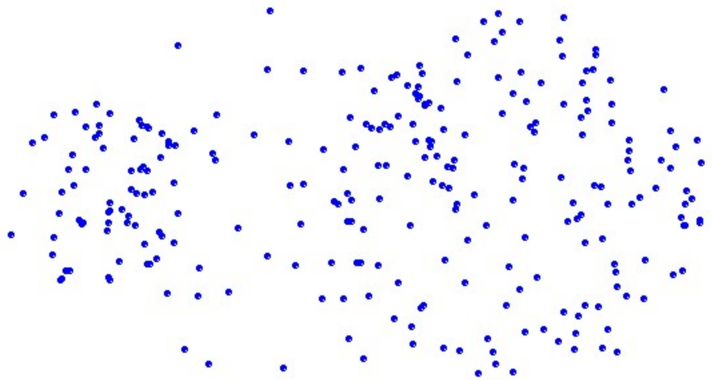


Two Clusters

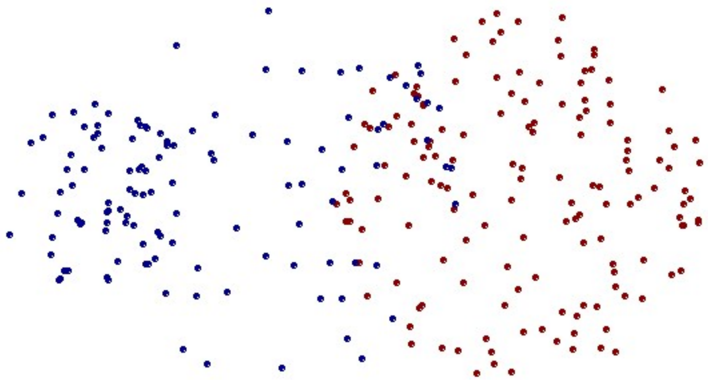
- Can handle non-elliptical shapes

Good for contiguity-based clustering

# Limitations of MIN



Original Points



Two Clusters

• Sensitive to noise and outliers

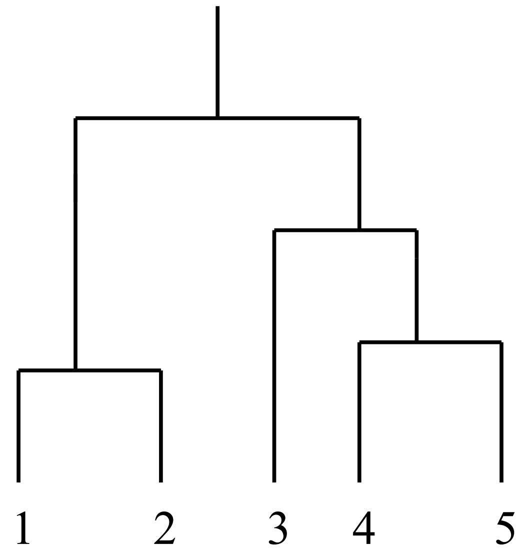
## Cluster Similarity: MAX or Complete Linkage

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by all pairs of points in the two clusters

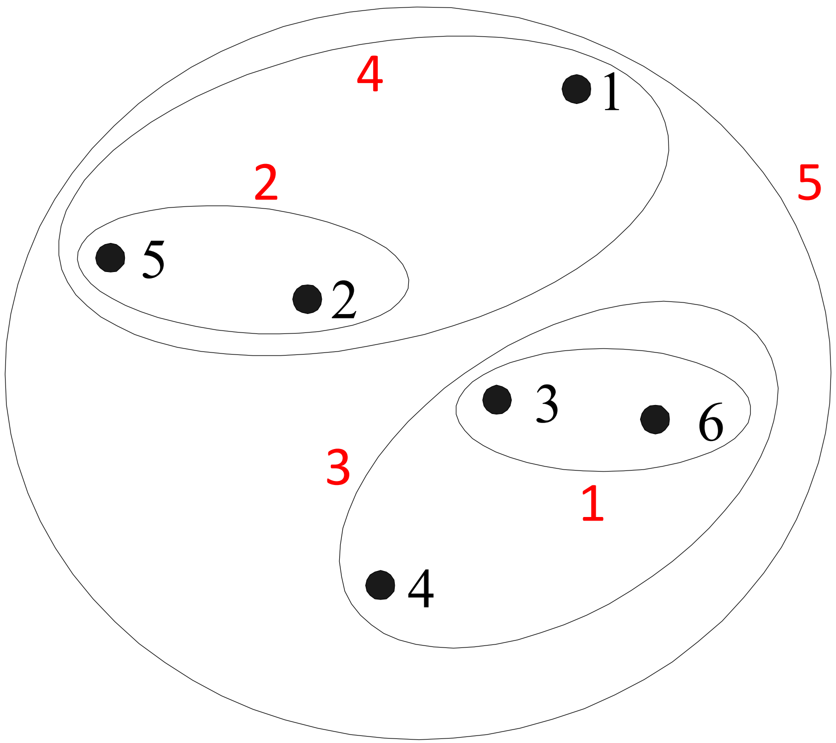
\- 两个聚类的相似性是基于不同聚类中两个最不相似（最遥远）的点

\- 由两个聚类中的所有点对确定

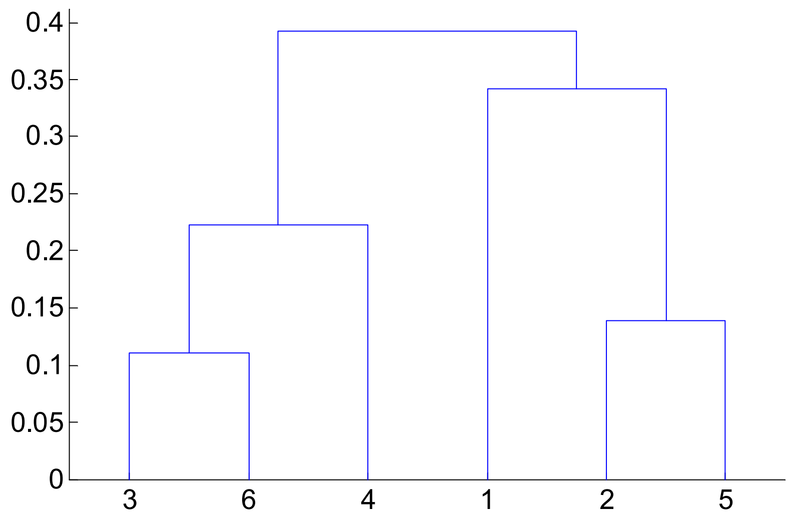
| | I1 | I2 | I3 | I4 | I5 |
|---|---|---|---|---|---|
| I1 | 1.00 | 0.90 | 0.10 | 0.65 | 0.20 |
| I2 | 0.90 | 1.00 | 0.70 | 0.60 | 0.50 |
| I3 | 0.10 | 0.70 | 1.00 | 0.40 | 0.30 |
| I4 | 0.65 | 0.60 | 0.40 | 1.00 | 0.80 |
| I5 | 0.20 | 0.50 | 0.30 | 0.80 | 1.00 |



# Hierarchical Clustering: MAX

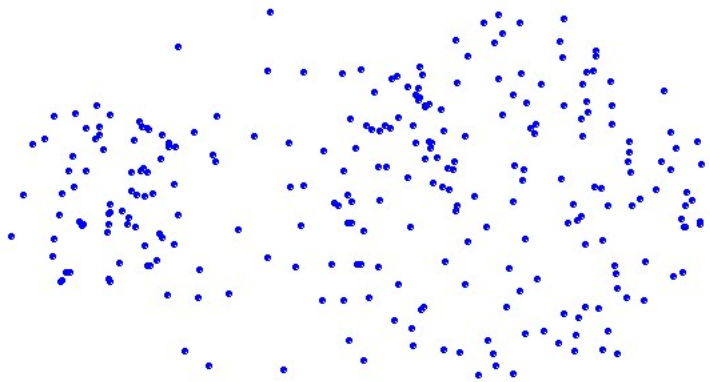



Nested Clusters

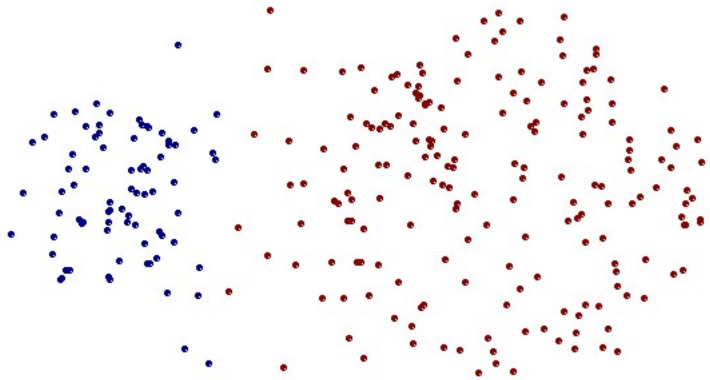



Dendrogram

# Strength of MAX



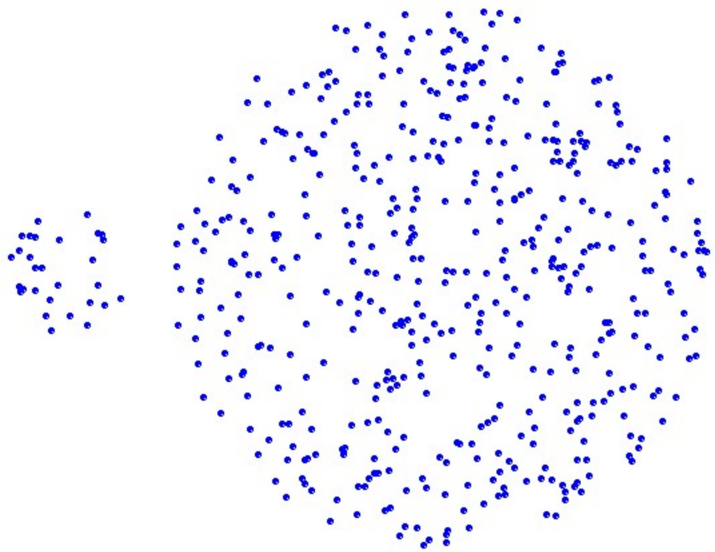
Original Points



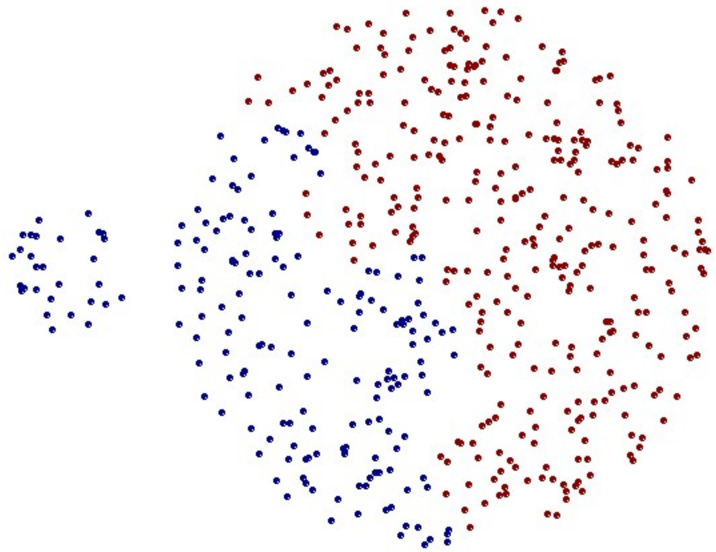
Two Clusters

- Less sensitive to noise and outliers

# Limitations of MAX



Original Points



Two Clusters

- Tends to break large clusters
- Biased towards **equal** globular clusters

# Cluster Similarity: Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

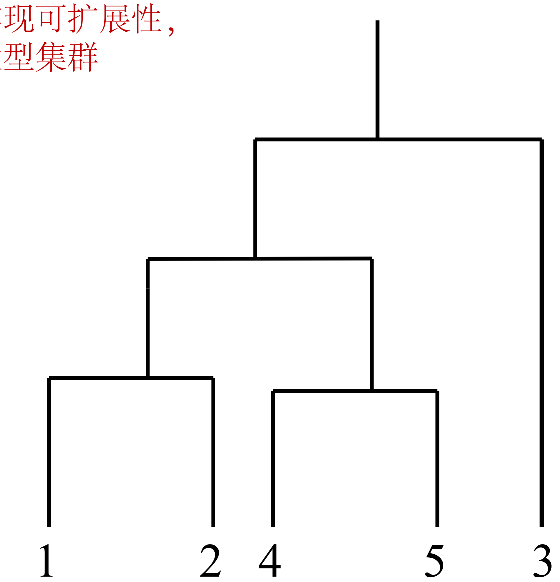
两个聚类的接近度是两个聚类中各点之间成对接近度的平均值。

$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum\limits_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

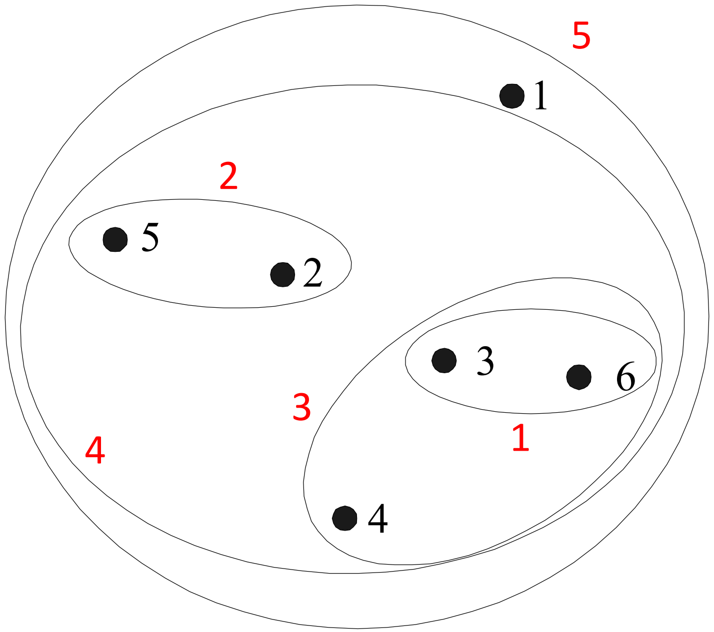
- Need to use average connectivity for scalability since total proximity favors large clusters

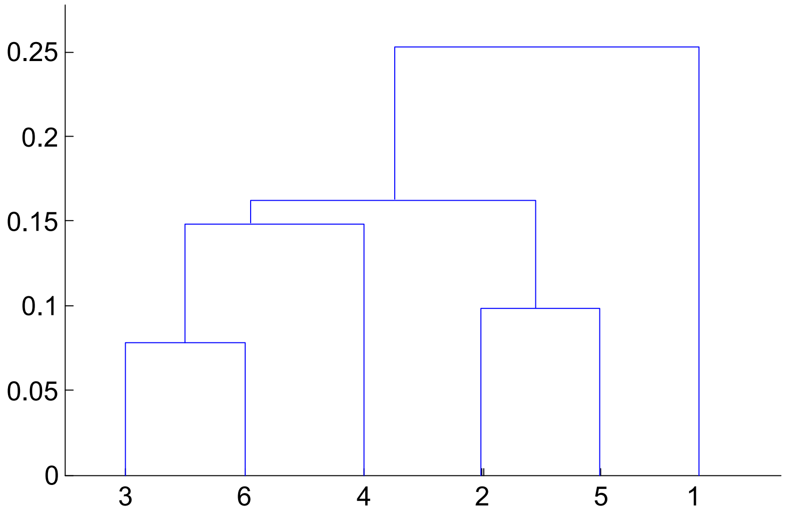
需要使用平均连接性来实现可扩展性，因为总的接近性有利于大型集群

| | I1 | I2 | I3 | I4 | I5 |
|---|---|---|---|---|---|
| I1 | 1.00 | 0.90 | 0.10 | 0.65 | 0.20 |
| I2 | 0.90 | 1.00 | 0.70 | 0.60 | 0.50 |
| I3 | 0.10 | 0.70 | 1.00 | 0.40 | 0.30 |
| I4 | 0.65 | 0.60 | 0.40 | 1.00 | 0.80 |
| I5 | 0.20 | 0.50 | 0.30 | 0.80 | 1.00 |



# Hierarchical Clustering: Group Average




Nested Clusters

Dendrogram

# Hierarchical Clustering: Group Average

- Strengths
  - Less sensitive to noise and outliers

- 优势
- 对噪音和异常值不那么敏感
- 限制因素
- 对球状星团有偏见

- Limitations
  - Biased towards globular clusters

# Example

Six data points are

$$x_1 = \begin{pmatrix} 1.5 \\ 4 \end{pmatrix},\ x_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix},\ x_3 = \begin{pmatrix} 1 \\ -1 \end{pmatrix},\ x_4 = \begin{pmatrix} 0.5 \\ -2 \end{pmatrix},\ x_5 = \begin{pmatrix} -1 \\ 1 \end{pmatrix},\ x_6 = \begin{pmatrix} 2 \\ -1.5 \end{pmatrix}$$

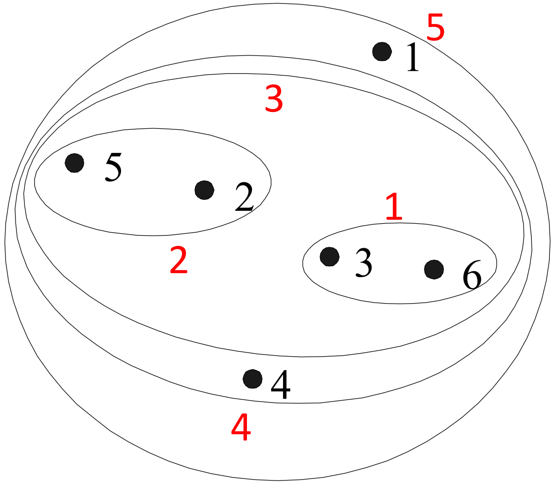
Please list the nested clusters by using hierarchical clustering based on MIN, MAX, and Group Average methods for inter-cluster Euclidean distance measure respectively. Note that, for each method, list all the intermediate Euclidean distance matrices.

5 minutes to try!

$$x_5 = \begin{pmatrix} -1 \\ 0.9 \end{pmatrix}$$

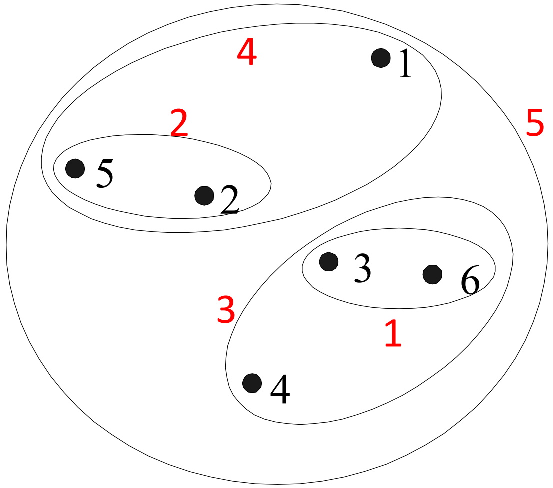
$$x_4 = \begin{pmatrix} 0.5 \\ -2.1 \end{pmatrix}$$

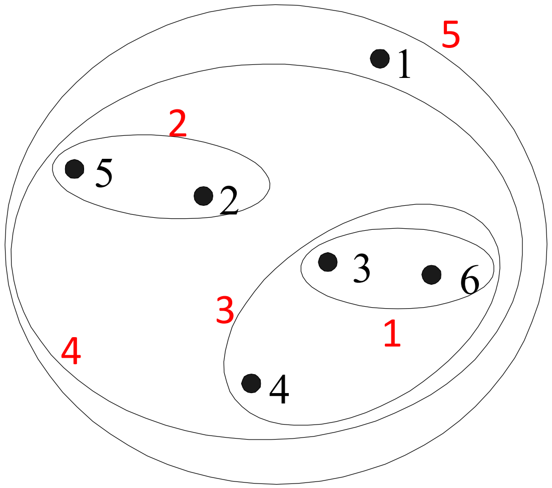
# Hierarchical Clustering: Comparison




MIN

MAX




Group Average

## Hierarchical Clustering: Time and Space requirements

- $O(N^2)$ space since it uses the proximity matrix.
  - N is the number of points.

- $O(N^3)$ time in many cases
  - There are N steps and at each step the size, $N^2$, proximity matrix must be updated and searched
  - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

- O(N2)空间，因为它使用接近矩阵。
- N是点的数量。
- 在许多情况下是O(N3)时间
- 有N个步骤，每一步都必须更新和搜索大小为N2的接近矩阵
- 对于某些方法，复杂性可以减少到O(N2 log(N) )时间

# Hierarchical Clustering: Problems and Limitations

- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters
  - Breaking large clusters

- 没有目标函数被直接最小化
- 不同的方案都有以下一个或多个问题。
- 对噪声和异常值的敏感性
- 难以处理不同大小的聚类
- 打破大型集群

# Measures of Cluster Validity

- 应用于判断聚类有效性各方面的数字措施，可分为以下两种类型。
- 外部指数。用于衡量聚类标签与外部提供的类标签的匹配程度。
- 熵
- 以主要类标签作为聚类标签的准确度
- 内部指数。 用来衡量聚类结构的好坏，而不考虑外部信息。
- 平方误差之和（SSE）

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following **two** types.
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
    - Accuracy with major class label as cluster label
  - Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)

# Other Trends

- Clustering with feature extraction
  - Ye at al. *Discriminative K-means* for Clustering, NIPS'07.

- Robust clustering with outliers
  - Liu et al. *Robust Subspace Segmentation by Low-Rank Representation*, ICML'10.

- Deterministic/convex clustering for given *K*